UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Mohar

Varnost v internetu stvari

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matej Mohar

Varnost v internetu stvari

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Mojca Ciglarič

Ljubljana, 2017

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo: Varnost v internetu stvari

Tematika naloge:

Preučite, na kakšne sisteme mislimo, kadar govorimo o internetu stvari. Identificirajte najpomembnejša področja in tipične primere uporabe. Nato naredite kratek pregled področja varnosti in navedite, katere od ranljivosti in tveganj opažamo v sistemih, ki sodijo v internet stvari. Varnostne pomanjkljivosti opišite na konkretnih primerih. Navedite mehanizme, orodja in ukrepe, s katerimi bi lahko dvignili varnostni nivo v internetu stvari in pojasnite, zakaj. Na primeru lastne aplikacije interneta stvari izpostavite varnostne pomanjkljivosti in jih odpravite s pomočjo prej predstavljenih ukrepov. Rešitev kritično ovrednotite in komentirajte, kdaj je varnostni nivo interneta stvari primeren.

Zahvaljujem se staršem in mentorici za podporo pri izdelavi diplomskega dela ter podjetju Vibor d.o.o., kjer so mi kljub delu omogočili opraviti vse opravke v zvezi z diplomskim delom.

Kazalo

Kazalo slik

Seznam uporabljenih kratic

kratica	angleško	slovensko
IDE	Integrated development environment	Integrirano razvojno okolje
IOT	Internet of Things	Internet stvari
SMS	Short message service	Sistem kratkih sporočil
M2M	Machine to Machine	Naprava z napravo
DDOS	Distributed Denail of Service	Porazdeljena zavrnitev storitve
TCP/IP	Transmission control protocol / Internet protocol	Protokol za nadzor prenosa / Interneti protokol

Povzetek

Naslov: Varnost v internetu stvari

Cilj diplomske naloge je pregled literature na področju interneta stvari in izdelava mobilne aplikacije za prikaz delovanja prenosa podatkov iz mobilnega telefona na spletni strežnik. Najprej so bili pregledani raznorazni članki na temo varnosti, najdenih je bilo nekaj primerov napadov ter njihovih posledic. V nadaljevanju so opisani principi na katere moramo paziti pri razvijanju sistemov, ki bodo vključeni v internet stvari. Ti principi sicer najbolj veljajo za proizvajalce strojne opreme, a vendar ima tudi vsaka strojna oprema svojo programsko podporo, ki je ranljiva na napade. Na koncu pa je še narejena aplikacija, ki ima poudarek na varnosti tako mobilnega dela kot tudi spletnega strežnika.

Ključne besede: internet stvari, varnost, Android, Xamarin, C#

Abstract

Title: Security in internet of things

The aim of this thesis is a review of a literature and making mobile application to show how data is transferred from mobile phone to web server. Firstly, lots of articles about internet of things security were examined, and search results found few examples of attacks on internet of things and their consequences. After that, there are described some principles, which must be considered and to which we must pay attention when developing systems that will be included in internet of things. These principles mostly apply to hardware manufacturers, yet every hardware device has some kind of software which may be vulnerable to attacks. At the end, there is concept of application that has focus on safety on mobile as well as on web server.

Keywords: Internet of things, security, Android, Xamarain, C#

1. Uvod

Internet stvari je v zadnjem času postal pomemben del našega vsakodnevnega življenja. Gotovo prav vsak izmed nas uporablja vsaj en mobilni telefon in računalnik, če že ne druge naprave povezane v internet. Mogoče se niti ne zavedamo da uporabljamo napravo ki spada v internet stvari. Recimo novi avtomobili. Nekateri avtomobilski proizvajalci [1] jih že označujejo za računalnike na kolesih in če spremljamo avtomobilske sejme, kot je sejem v Genevi in CES, sejem zabavne elektronike, opazimo trend, da je že dosti avtomobilskih proizvajalcev prisotnih tudi na drugem.

Mogoče se nam to zdi v redu, da bodo avtomobili vedno bolj povezani v internet in nas bodo spremljali, a ne zavedamo se nevarnosti ki jo to prinaša. Prav tako to velja za veliko število drugih, manj očitnih naprav povezanih v splet, ki jih imamo okoli nas. Zato je pomembno, da smo ljudje obveščeni in da se zavedamo kaj se dogaja tudi pod »pokrovom« sive škatlice, ki nam v stanovanju upravlja s temperaturo.

Kaj se lahko zgodi, če proizvajalec naprave ni predvidel vseh nevarnosti in so bile odkrite kasneje, ko je bila naprava že v proizvodnji? A vemo? Večina verjetno ne, nekateri pa se že zavedajo in poskušajo določiti principe po katerih bi morali delovati proizvajalci naprav in tako na podlagi teh principov bi bile naprave bolj varne.

V diplomskem delu si bomo najprej pogledali kaj sploh je internet stvari in se sprehodili skozi njegovo zgodovino vse do današnjih dni ter pogledali kaj prinaša prihodnost. Pogledali si bomo pa tudi eno zelo pomembnih poglavji v internetu stvari in sicer je to poglavje o varnosti.

Varnost je zelo pomemben člen, saj nas nekateri [2] opozarjajo, da je vsaka naprava povezana v splet ranljiva na napade, zaradi tega je pomembno, da se poskusimo čimbolj zaščititi. In sebe ter svoje podatke lahko zaščitimo tako, da začnemo razmišljati o varnostni naprav, ki jih uporabljamo.

Navedenih bo nekaj primerov napadov ter kako so se proizvajalci nanje odzvali, na koncu pa so predstavljeni tudi nekateri principi, ki bi se jih proizvajalci naprav in programske opreme morali držati za varnejšo uporabo.

Nazadnje pa je predstavljena aplikacija, ki prikazuje še kako lahko kot programerji z malo truda izdelajo dokaj varno mobilno aplikacijo, ki prenaša podatke med mobilnim telefonom uporabnika ter spletnim strežnikom, seveda na čimbolj varen način.

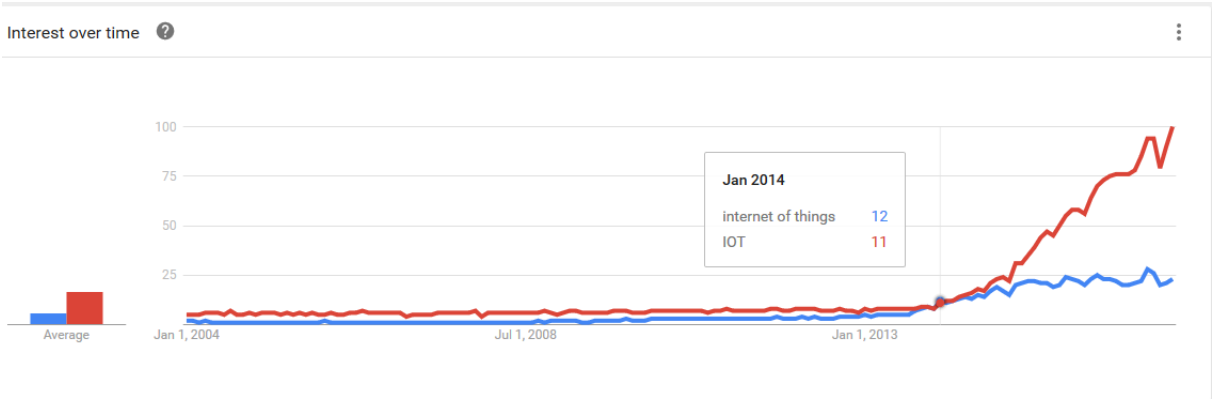
2. Kaj je internet stvari

Internet stvari ali »Internet of Things«, kot se temu reče v angleškem jeziku – iz tega pa je nastala tudi zelo znana kratica IOT, pomeni mnogo v internet povezanih naprav, ki s pomočjo programske opreme in raznih senzorjev zbirajo velike količine podatkov ter si te podatke lahko med seboj tudi izmenjujejo.

Ideja interneta stvari je, da imajo naprave, ki so povezane v internet, možnost komuniciranja z uporabnikom, z raznimi drugimi aplikacijami ter konec koncev tudi druga z drugo. Kot primer lahko naprava z nami komunicira preko SMS sporočil, preko elektronske pošte ali pa preko katerokoli drugega komunikacijskega kanala. Isto velja v obratni smeri, saj naprava lahko deluje tako, da se odzove na naš ukaz ki smo ji ga poslali. Naprava, ki je del IOT, si lahko izmenjuje podatke z drugo napravo in se tako zaveda vseh parametrov v svojem sistemu. Da te na novo pridobljene informacije čim bolje izkoristi, jih lahko pošilja v določeno podatkovno bazo. Tam jih za to namenjena programska oprema analizira, ter poskuša iz njih izluščiti čimbolj uporabne informacije. Ker je takih podatkov lahko veliko, se takim podatkovnim bazam v angleškem jeziku reče Big data.

2.1 Zgodovina interneta stvari

Še ne dolgo nazaj marsikdo ni vedel o obstoju interneta stvari, danes pa se to spreminja. To trditev pa potrjuje tudi Slika 1: Prikaz trenda za pojem Intenet of Things in njegovo kratico., ki prikazuje število iskanj za internet stvari. Vedno več se o njem govori in vedno več ljudi se zanima za ta koncept. Podjetja tudi vedno več vlagajo v razvoj interneta stvari, zato se na prvi pogled mogoče zdi da je tak koncept nova stvar, stvar tega desetletja, kar pa ni res.



Slika 1: Prikaz trenda za pojem Intenet of Things in njegovo kratico. [3]

Slika 1 prikazuje popularnost angleških izrazov »Internet of Things« ter kratice »IOT«. Hitro opazimo, da se popularnost od začetka beleženja leta 2004, pa do nekje konca leta 2013 nič kaj dosti ne spreminja, ampak je kar konstantno nizka. Nekje v začetku leta 2014 pa se je zgodil preobrat in zanimanje za internet stvari sunkovito narašča vse do danes. Kljub temu, da se do leta 2014 zanimanje za angleška pojma internet stvari ni večalo, pa so se prvi zametki začeli kazati že konec osemdesetih let prejšnjega stoletja. Takrat sta John Romkey ter Simon Hackett v internet povezala prvi toaster [4], še pred tem pa je bilo nekaj zelo pomembnih mejnikov ter omemb in predvidevanj znanih raziskovalcev, ki so delali na tem področju. Katera leta in dogodki so bil prelomni in so tudi zgodovinsko pomembni za razvoj današnjega interneta, pa je napisano v spodnjem seznamu.

- **1844:** Samuel Morse za 1. Maj prvič demonstrira prenos informacije na 61km dolgi razdalji, med Washington, D.C. ter Baltimorom. Le nekaj tednov kasneje se povezava tudi uradno odpre in Morse pošlje znane besede iz Biblije, »What hath God wrought«. [5]

- **1926:** Januarja tega leta je imel Nikola Tesla intervju z revijo Colliers, v katerem je povedal veliko napovedi. V eni izmed napovedi je povedal, da bo enkrat v prihodnosti, ko bo brezžična tehnologija v polni uporabi cel svet postal povezan ter da se bo cel svet obnašal kot eni možgani. Med seboj bomo povezani in slišali ter videli se bomo med seboj, ne glede na razdaljo med nami. Dejal je, da bomo imeli tako napravo v svojem žepu. Danes temu pravimo mobilni telefon. Tesla je povedal *»When wireless is perfectly applied the whole earth will be converted into a huge brain, which in fact it is, all things being particles of a real and rhythmic whole. We shall be able to communicate with one another instantly, irrespective of distance. Not only this, but through television and telephony we shall see and hear one another as perfectly as though we were face to face, despite intervening distances of thousands of miles; and*

the instruments through which we shall be able to do his will be amazingly simple compared with our present telephone. A man will be able to carry one in his vest pocket.« [6]

- **1964:** Marshall McLuhan je v svoji knjigi Understanding Media dejal: *»....by means of electric media, we set up a dynamic by which all previous technologies -- including cities -- will be translated into information systems«,* kar se dogaja danes z uporabo interneta stvari – pametna mesta se razvijajo.

- **1969:** V Združenih državah Amerike skupina iz podjetja BBN ter Agencije za napredne obrambne analize (DARPA) razvijeta prvo delujoče omrežje z imenom ARPANET (ki je kratica za Advanced Research Projects Agency Network), ki je osnova za današnji Internet. 29. Oktobra je bilo poslano tudi prvo sporočilo, ki je vsebovalo besedo »LOGIN«. [7]

- **1974:** Prvič se pojavi TCP/IP protokol, ki je danes osnova za večino komunikacije na internetu.

- **1990:** Tega leta se smatra da je bila narejena prva naprava, ki bi jo lahko uvrstili v internet stvari. John Romkey ter Simon Hackett sta na Interop sejmu predstavila v internet povezan toaster, ki je postal hit takratnega sejma. Slika 2 prikazuje Simona, ki demonstrira delovanje toasterja. V internet je bil povezan s TCP/IP protokolom, kontrolirali pa so ga z SNMP/MIB protokolom (Simple Networking Management Protocol Management Information Base). Imel je le en ukaz in sicer to, da je toaster vključil, kako zapečen bo kruh, pa je kontroliral s tem koliko časa je imel napajanje. Leto kasneje so dodali še avtomatsko roko, ki je pobrala košček kruha in ga dala v toaster. [4]

Slika 2: Simon Hackett demonstrira prvi internetni toaster [4]

- **1991:** Tim Berners v CERNu vzpostavi prvo spletno stran.

- **1999:** Prvič se pojavi nam danes dobro poznan izraz »Internet of Things«. Prvi, ki ga je skoval pa je Kevin Ashton. Kevin se je tudi poglobil v razvoj RFID čipov z namenom zamenjave črtnih kod. Istega leta je Neil Gross za članek Buisness Week [8] dejal, da bomo v naslednjem stoletju priča temu, da bo zemlja dobila svojo »kožo«. Le ta bo sestavljena iz milijone majhnih senzorjev, ki bodo zajemali temperaturo, onesnaženost, zvok, sliko, EKG. Nadzirali bodo živalske vrste, imeli jih bomo na ladjah, avtocestah, vozilih… Dejal je točno to, kar je danes internet stvari.

- **2000:** LG je predstavil prvi hladilnik povezan v internet.

- **2005:** To je bilo veliko leto za internet stvari, saj je organizacija Združenih Narodov prvič omenila IOT v svojem International Telecommunications Union poročilu. [9] V poročilu so dejali, da je prav internet stvari dodal novo dimenzijo k komunikacijskim tehnologijam, saj imamo sedaj povezavo kadarkoli in kjerkoli. Nekaj let kasneje je bila v Evropski Uniji prva konferenca na temo interneta stvari.

- **2008-2009:** Glede na Cisco Internet Buisness Solutions Group je bil internet stvari »rojen« v teh letih, saj je bilo prvič v splet povezanih več naprav kot ljudi. Glede na naraščajočo ponudno mobilnih telefonov, tablic in ostalih povezanih stvari je bilo leta 2010 v internet povezanih že več kot 12,5 milijard naprav, ljudi pa je bilo takrat 6,8 milijard.

- **2010:** Google je predstavil projekt za avtonomno vožnjo avtomobila.

- **2011:** Zagnan je bil protokol IPv6, ki omogoča 2^{128} internetnih naslovov. Ta protokol je za razvoj interneta stvari zelo pomembno, saj je IPv4 naslovov hitro primanjkovalo. Z IPv6 protokolom pa je naslovov dovolj.

2.2 Kje smo danes in kakšna je prihodnost

Kar se je leta 1999, ko je nastal izraz »Internet of Things« zdela še skoraj da ne znanstvena fantastika, je danes realnost. Svet je vse bolj povezan in vse bolj smo navajeni srečevati tako imenovane pametne naprave. Nekatere smo že tako navajeni, ali pa so tako dobro integrirane v vsakdanje življenje, da jih sploh ne opazimo. Število naprav povezanih v internet se danes šteje v milijarde, napoved za prihodnost pa je zelo obetajoča.

V kategorijo internet stvari spadajo vse naprave, ki so kakorkoli povezane v internetno omrežje in zmožne komunicirati z ljudmi, programsko opremo ali pa ostalimi napravami. Z internetom stvari se danes srečujemo vsak dan, med najbolj znane pa štejemo:

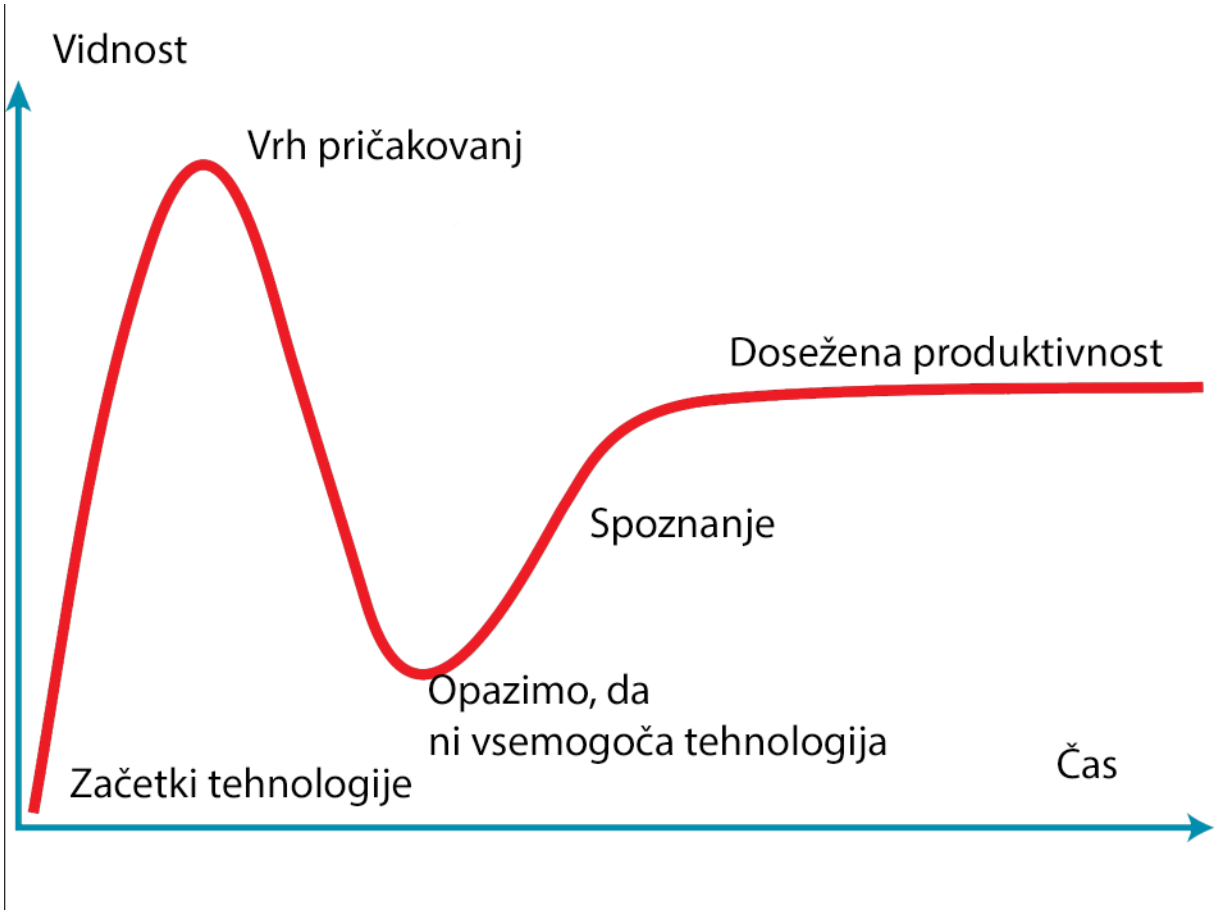
- Hladilne, ventilacijske in klimatske naprave, ki so že povezane v Internet. Z njimi se lahko na primer upravlja kar z aplikacijo na mobilnem telefonu.

- Avtomobili, povezani v internet. Tudi na tem področju se dogaja veliko, saj je vse več novih avtomobilov stalno povezanih v internet preko mobilnega omrežja. Taki avtomobili lahko ob prometni nesreči sami sporočijo lokacijo ter stanje v prometni center, kljub temu da je človek lahko nezavesten. Lahko pa se tudi posodabljajo, diagnosticirajo ter tudi upravljajo na daljavo. Primer takega avtomobila je Tesla Model S. [10]

- Pametni mobilni telefoni, kar ni nič presenetljivega, saj zmorejo danes skoraj vse kar se pričakuje od navadnega računalnika.

- Skoraj vsaka nova televizija ima danes že možnost povezave na splet. Nanjo priklopimo miško, tipkovnico, pa nam služi kot računalnik. [11]

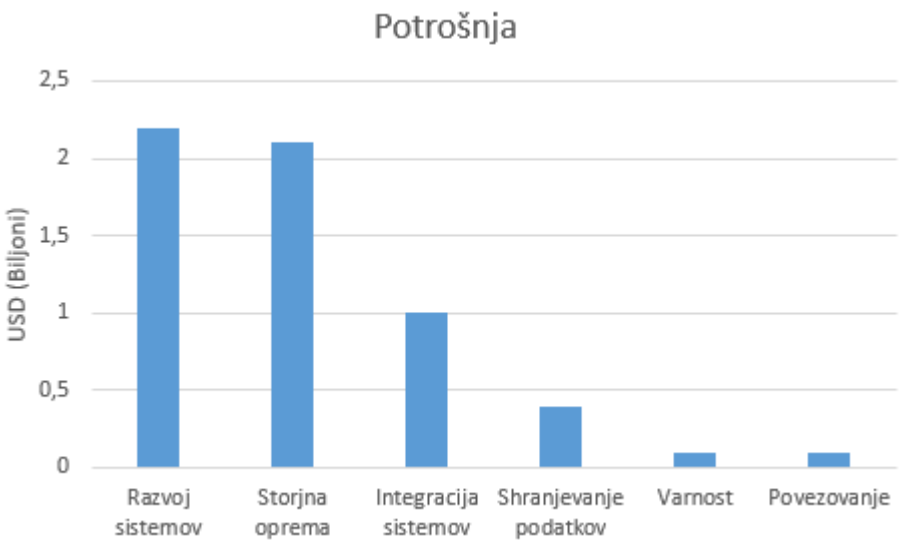
Prihodnost je vsekakor svetla in če za primer vzamemo Gartnerjevo [12] raziskavo o prihajajočih tehnologijah, opazimo da je v raziskavi internet stvari malo pod vrhom krivulje prihajajočih tehnologij.



Slika 3: Krivulja pričakovanj novih tehnologij

Kaj naj pravzaprav taka krivulja kot jo prikazuje Slika 3 pove? Internet stvari se nahaja v drugem delu krivulje, tik pod vrhom. To pomeni da je tehnologija že dovolj zrela, da so mnogi že to tehnologijo uporabili ter z njo tudi uspeli. Mnogi pa poskusili, ampak jim ni uspelo. Glede na krivuljo bo zanimanje za internet stvari upadlo, ampak bodo potem podjetja in posamezniki spoznali še boljše ideje kako bi jo lahko uporabili in jo tudi bodo. Takrat se bo začela masovna uporaba in internet stvari bo v zadnjem delu te krivulje. Gartner raziskava temu pripisuje 5 do 10 let.

Danes je glede na raziskave v internet povezanih približno 10 do 15 milijard naprav in senzorjev. V prihodnjih treh letih, torej do leta 2020 pa raziskave kažejo, da bo v internet povezanih že med 25-50 milijard naprav. Vloženega bo tudi veliko denarja in od interneta stvari se pričakuje tudi veliko povratnih sredstev.

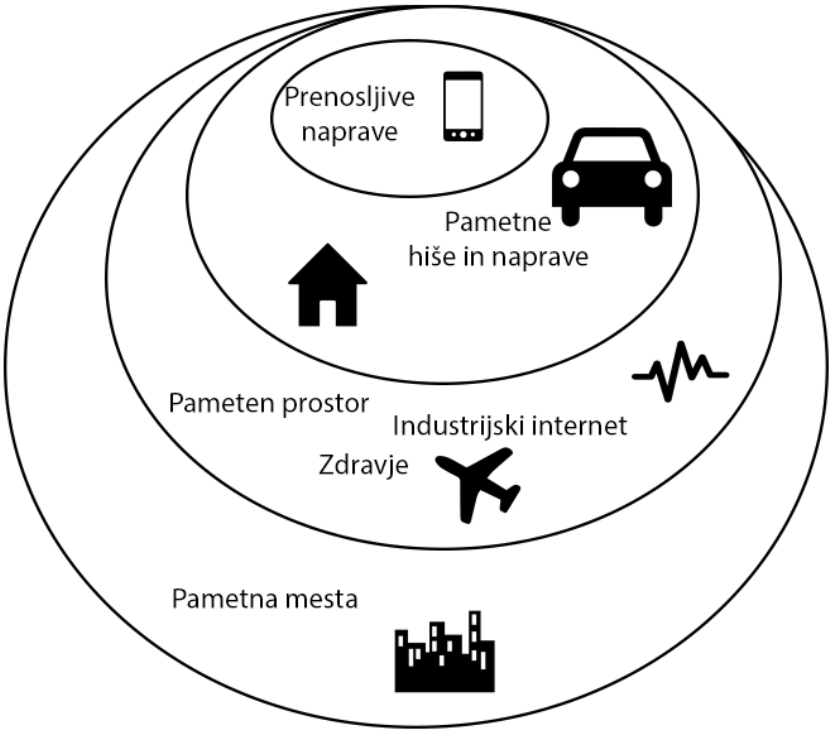


Slika 4: Kje se bo v naslednjih letih porabilo največ denarja, povzeto po članku Buisiness Insider [13].

Kot prikazuje Slika 4, se predvideva da se bo skupaj do leta 2020 porabilo 6 bilijonov Ameriških dolarjev. Zanimiva stvar tega je, da se bo za varnost IOT naprav namenilo le deček tega, največ pa za razvoj nove strojne opreme ter aplikacij. [13]

2.3 Primeri uporabe interneta stvari

Internet stvari – oz. razne med seboj povezane senzorje in naprave, se uporablja skoraj da ne povsod. Ljudje jih nosimo vedno s sabo (pametne ure, mobilni telefoni), imamo jih doma in z njihovo pomočjo preko mobilnega telefona upravljamo klimatsko napravo, prižgemo luč, zapremo garažna vrata, odpremo okno, ugasnemo računalnik in konec koncev tudi preverimo če je pralni stroj že opravil svoje delo. Internet stvari pa se ne ustavi tukaj, ampak gre dlje. Uporabljajo ga letala, ki beležijo podatke, kmetje, ko spremljajo zdravje in lokacijo svoje živine, kljub temu da se živina pase na pašniku kilometre proč. V gostilni lahko naročimo preko tablice ali pa ko poštar prinese pismo, se podpišemo na terminal, ne na papir. Vedno več se uveljavlja tudi koncept pametnih mest. [14]



Slika 5: Kategorije IOT

Slika 5 prikazuje kategorije za internet stvari, ki so navedeni in podrobno opisani v spodnjih podpoglavjih.

2.3.1 Prenosljive naprave

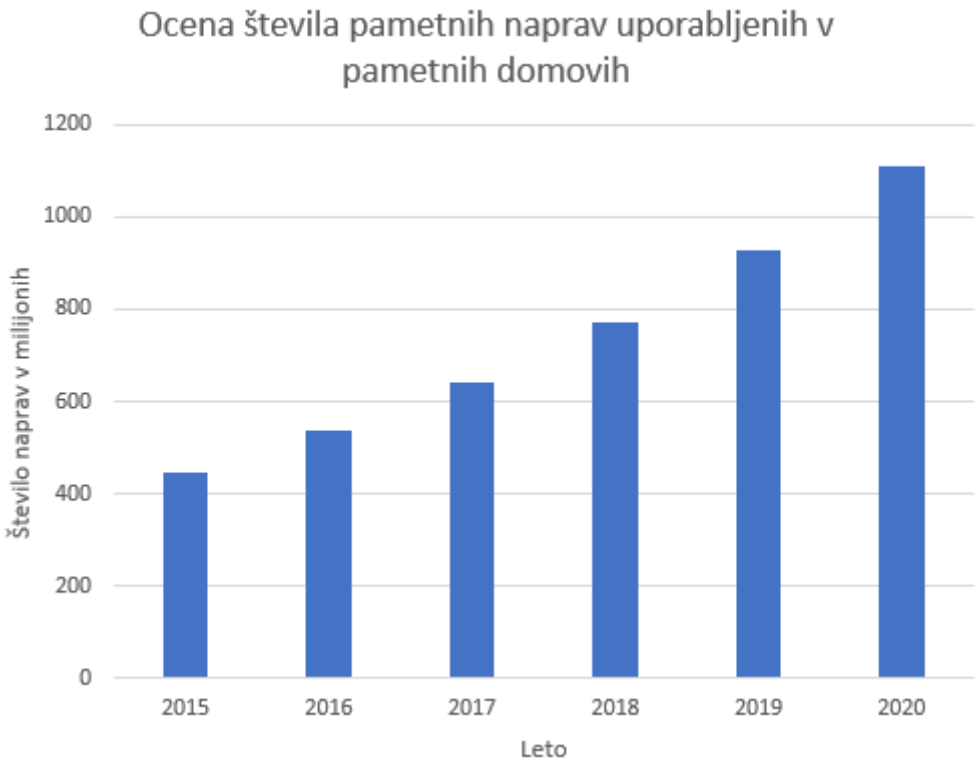
To so naprave, ki jih večino časa nosimo s sabo in so točno temu tudi namenjene. Cilj takih naprav je vsekakor izboljšati naša življenja. Primer takih naprav je pameten telefon ali pa pametna ura, kot je na primer Apple Watch, ki jo prikazuje tudi Slika 6.

Slika 6: Pametna ura Apple

Prenosljive naprave nam služijo predvsem za to, da lahko vsak spremlja svoje dosežke. Veliko se jih uporablja za spremljanje treningov in svoje telesne pripravljenosti. Pametna ura pa lahko na primer deluje kot digitalni trener, lahko analizira naše športne dosežke in vse to nalaga v oblak.

2.3.2 Pametne hiše in naprave

Pametne hiše so opremljene s senzorji, ki spremljajo bivalne pogoje ter glede na naše počutje spreminjajo temperaturo, svetlobo, zvok… Število pametnih hiš in senzorjev se zelo hitro povečuje, saj nam take naprave ne samo izboljšajo počutje, ampak z njimi lahko tudi privarčujemo na stroških. Predvideno je da bo na tržišču leta 2020 preko milijarda (Slika 7) naprav in senzorjev za pametne domove.



Slika 7: Število naprav za pametne hiše

Nekaj konkretnih primerov uporabe:

- Pametno upravljanje z energijo: Če imamo v hiši pameten termostat, ga lahko naučimo da se hiša avtomatsko ogreje na nam prijetno temperaturo ko se vrnemo iz službe ter da ponoči ko spimo stanovanje malo ohladi, zjutraj ko se zbudimo pa zopet ogreje

- Varnost: Vrata se zaklepajo glede na našo pozicijo, lahko jih zaklenemo s pomočjo aplikacije na mobilnem telefonu ali preko računalnika. Tudi iz drugega konca sveta.

2.3.3 Pameten prostor

V tem segmentu imamo pametne naprave za nadziranje farm, trgovin, polj. V tem segmentu je meni zelo pomembno poljedelstvo in živinoreja. Glede na raziskave »UN Food and Agriculture Organization« bomo v letu 2050 morali pridelati 70% več hrane kot jo pridelamo sedaj. [15] In to bomo storili veliko lažje, če bomo v kmetijstvo vpeljali internet stvari. Se pa internet stvari v kmetijstvu že uporablja:

- Razni senzorji postavljeni čez polja nadzirajo vlažnost, temperaturo in ostale spremenljivke okolja. Senzorji imajo lahko dostop do vremenske napovedi in z vsemi zbranimi podatki izvajajo statistične napovedi za število in kvaliteto pridelka.

- Veliko vlogo pri kmetijstvu ima tudi vizualni del. Za imeti nadzor nad poljem pa se veliko uporablja majhne helikopterje na daljinsko upravljanje.

- Podobno kot v prometu, kjer se intenzivno razvijajo avtonomni avtomobili, se v kmetijstvu stremi k temu, da bi imeli avtonomne traktorje in ostalo mehanizacijo.

2.3.4 Zdravje

Vsak dan mi vsi generiramo ogromno podatkov. Nekateri so očitni – kot so na primer telefonski pogovori, GPS pozicija, podatki iz raznih športnih ur in podobno. Nekateri pa niso toliko očitni, a vseeno pripomorejo k razumevanju posameznika. Na primer podatki iz pametnega termostata ali pa senzorjev za vrata. Problem tega je, da je tako pridobljene podatke danes zelo težko zbrati na enem mestu in v formatu primernem za nadaljnjo obdelavo, saj danes še ne obstaja centraliziran sistem, ki bi to omogočal. Vsi ti podatki pa nam lahko omogočijo da preko raznoraznih analiz spoznamo določene navade ljudi in mogoče predhodno odkrijemo znake bolezni ali drugih tveganj.

Pri uporabi interneta stvari v zdravstvene namene pa smo ljudje veliko bolj previdni, saj so lahko posledice nepravilnega delovanja zelo hude – tudi smrtno nevarne. Varnost je vsekakor pomembna v celotnem IOT sistemu, ampak še toliko bolj, ko dodamo zraven pacientove privatne informacije. Na svetu bo že bili primeri ko so napadalci onesposobili informacijski sistem določene bolnišnice in tako zdravniki niso mogli dostopati do pacientovih podatkov. [16] Zato je varnost v zdravju še posebej pomembna.

2.3.5 Industrijski internet

Zelo podobno kot pri pametnem prostoru, se tudi tukaj uporabljajo senzorji, katerih beleženi podatki se uporabljajo za boljše razumevanje procesa proizvodnje. V proizvodnji je zelo pomembno da se naprave pogovarjajo med sabo, za to se največ uporablja M2M protokol (Machine 2 Machine). Dober primer je pridobivanje energije iz obnovljivih virov, kjer je treba planirati porabo, pa tudi čimbolj v naprej vedeti kdaj bo stroj potreben vzdrževanja. Sicer pa pod industrijski internet spada vse od organiziranja in vodenja avtobusov, vlakov, rudarjenja in podobnih panog.

2.3.6 Pametna mesta

Pametna mesta so mesta, kjer so za določene primere postavljeni senzorji in naprave, ki reagirajo na podatke teh senzorjev. Največkrat je bistvo pametnih mest izboljšanje kakovosti življenja prebivalcev in seveda tudi prihranek denarja.

Slika 8: Pametne luči v Amsterdamu

Dober primer je lahko Amsterdam [17], kjer jim je cilj zmanjšati promet, prihraniti energijo ter izboljšati varnost mesta. Za prihranek energije s senzorji zaznavajo število pešcev ter glede na pridobljene podatke prilagajajo intenzivnost svetlobe, ki jo oddajajo luči (Slika 8). Poleg tega pa tudi spremljajo promet v realnem času in te podatke oddajajo drugim, ti pa se lahko na podlagi teh podatkov odločijo za drugo pot od načrtovane. Tako se izognejo gneči. V pametno mesto lahko spada še:

- Zaznavanje prostih parkirnih mest in sporočanje tistim ki parkirno mesto iščejo
- Nadzor nad dogajanjem v mestu preko kamer in raznih senzorjev, da se lahko čimprej ukrepa ko pride do nevšečnih dogodkov kot so uporaba orožja, pretepi ali pa naravnih katastrof – poplave, potres

3. Napadi in varnost

Naprav ki se smatrajo kot sestavni del interneta stvari je danes veliko. Srečamo jih na vsakem koraku in v prejšnjem poglavju je opisanih nekaj primerov uporabe v realnem svetu. Veliko vprašanje pa je, ali je uporaba interneta stvari danes varna?

Ko se je začela ideja o internetu stvari, približno 20 let nazaj, si še noben ni mislil da nam bodo s pomočjo našega telefona lahko ukradli naše biometrične podatke (prstni odtis, zenica…), številko bančnega računa, našo pozicijo in še mnogo, mnogo več. [18] Danes vemo da se to lahko zgodi in počasi se začenjamo zavedati, da nam lahko napadalci ukradejo podatke tudi s pomočjo raznih napravic povezanih v internet – pa naj bo to pametna televizija, hladilnik, klimatska naprava ali pa opekač kruha. Če naprava ni pravilno zaščitena lahko napadalec dobi nadzor nad njo in mogoče tudi nad našim celim omrežjem. To mu omogoči, da naše omrežje in naprave izkorišča za morebitno nelegalno početje – DDoS napadi ali na primer gostovanje nelegalni vsebin. Z veliko gotovostjo lahko trdimo da je zasebnost pomembna veliki večini ljudem, zato je še toliko bolj pomembno da je internet stvari varen. Noben si namreč ne želi da bi ga popolni tujci gledali, poslušali ter spremljali vse njegove korake. Konec koncev bi lahko nekdo, ki dostopa do vseh takih informacij osebo tudi oškodoval. Pa se to dogaja?

3.1 Napadi in varnostne pomanjkljivosti

Zlorab naprav povezanih v internetno omrežje je veliko – vsak dan več. Del interneta stvari predstavljajo tudi take naprave ko so mali senzorji elektronika, ki se jo le namesti, nato pa se jo več ne posodablja in če ne nudimo zaščite na višjem nivoju, potem je naprava ranljiva. In ena naprava je dovolj, da nepooblaščene osebe dostopajo do našega omrežja. V naslednjih podpoglavjih je opisanih nekaj napadov na pametne naprave in rešitve, ki so jih izbrali proizvajalci naprav ali pa so bile predlagane s strani raziskovalcev.

3.1.1 Pametne ure

Velika večina pametnih ura ima vgrajen tudi žiroskop in senzorje premikanja in ker pametne ure nosimo skoraj ves čas, so le te zanimive za razne hekerske skupine ali posameznike. Ure pa nosimo tudi ko tipkamo po tipkovnici in vnašamo razna gesla ter osebne številke. Raziskovalci [19] so za ta namen pripravili aplikacijo za Samsung pametno uro, ki sledi premikanju zapestja in beleži podatke, ki jih beležita pospeškometer ter žiroskop. Aplikacijo so testirali in ko so zbrali podatke o premikanju ure, so podatke analizirali in sestavili 2D

model premika roke (ure). Primer tega, kako so ugibali katera tipka je pritisnjena je, da se za črko »T« z zapestjem pomaknemo dlje, kot če želimo pritisniti črko »F«. Razvita aplikacija pa še ni bila sposobna identificirati vseh pritisnjenih tipk, kot so na primer posebni znaki, številke ter ločila. Težavo je predstavljala tudi preslednica, poleg tega pa je lahko aplikacija zbirala podatke o tipkanju le tistih tipk, ki so bile pritisnjene z uporabo roke, na kateri je uporabnik nosil pametno uro.

Drugi primer pa je primer študenta, ki je prav tako razvil sistem za beleženje kaj tipkamo, s pomočjo pametne ure. [20] Ta aplikacija pa beleži tipkanje po numeričnem delu tipkovnice. Kar pomeni, da lahko beleži vse vnesene PIN številke na bankomatih ter gesla na mobilnih telefonih. Študentu Beltramelli, ki je razvil to aplikacijo, je uspelo doseči do 73% točnost.

V obeh navedenih primerih gre za podoben sistem, kjer so za identifikacijo pritisnjenih tipk razvili svojo aplikacijo. To pa lahko predstavlja potencialni problem, saj bi lahko nekdo razvil aplikacijo, ki bi bila na primer namenjena beleženju prehojene razdalje, vmes pa bi beležila še naše tipkanje. Uporabniki, ki bi namestili to aplikacijo se lahko nebi zavedali nevarnosti, ki jo predstavlja, saj bi se za beleženje prehojene razdalje uporabljalo enake senzorje kot za identifikacijo pritisnjenih tipk. Da zmanjšamo verjetnost takega napada, raziskovalci predlagajo, da bi bilo zelo užinkovito, če bi pametne ure zmanjšale frekvenco zaznavanja gibov. Večina pametnih ur gibanje zaznava do 200x v sekundi, če pa bi zaznavo gibanja zmanjšali na 15x v sekundi, bi bilo sledenje gibanju zapestja precej težje opravilo. Res pa je tudi, da ure ne bi več tako natančno beležile gibanja, kot smo ga vajeni do sedaj.

3.1.2 Srčni spodbujevalniki

Če so hekerji pri pametnih urah samo kradli podatke, so pri srčnih spodbujevalnikih posledice lahko precej hujše – tudi smrt. [21] Raziskovalcu Barnaby Jacku je uspelo demonstrirati, da lahko srčni spodbujevalnik spremeni tako, da pacientu zadane serijo 830 voltov močnih električnih šokov, ki so že smrtno nevarni. V video predstavitvi na BreakPoint varnostni konferenci je demonstriral, kako lahko z prenosnim računalnikom na 10 metrov razdalje pacientu zadane smrtno visok tok. Predstavitve sicer ni javno objavil, saj bi lahko iz nje izvedeli proizvajalca ter model ranljivega srčnega spodbujevalnika. Povedal je, da bi lahko brez večjih težav povzročil smrt pacienta v bližini, a njegov največji strah je bil ta, da bi lahko sprogramiral napravo tako, da bi vsaka naprava okužila sosednje naprave in tako bi lahko napadalec oškodoval veliko ljudi.

V Ameriki je FDA (US Food and Drug Administration) objavila poročilo in napotke [22] ki naj bi jih upoštevali proizvajalci medicinske opreme, torej med drugim tudi proizvajalci srčnih spodbujevalnikov. V poročilu povedo, da je veliko naprav povezanih v internetno

omrežje bolnišnic ter tudi internetno omrežje doma pri pacientih, kar sicer olajša delo beleženja zdravja bolnikov in izboljša kvaliteto oskrbe, a vendar predstavlja tudi veliko varnostno tveganje za internetne napade. V poročilu navedejo da je najboljša preventiva takih napadov to, da proizvajalci že zasnujejo napravo v mislih na čim večjo varnost, kasneje pa naj jo skozi cel življenjski cikel spremljajo in ob zaznanih nepravilnostih tudi posodobijo.

Problem takih naprav kot so srčni spodbujevalniki je, da ko jih podjetje enkrat spravi na trg, se jih pogosto ne posodablja več. Kar pomeni, da če nekdo odkrije luknjo v sistemu, je na svetu še vedno veliko uporabnikov ki uporabljajo to napravo, a na njihovih napravah luknja ostaja nezakrpana. Zaradi tega je tudi FDA izdala še eno priporočilo, in sicer, da ko proizvajalec naprave izve za napako, jo mora odpraviti v roku dveh mesecev. Dva meseca je sicer dolgo obdobje – pri mobilnih telefonih je ta čas ponavadi krajši – a vendar bolje luknjo odpraviti v dveh mesecih kot nikoli. Ker pa so to samo priporočila FDA in ne obvezujoče zahteve, je po mojem mnenju malo verjetno da se bodo proizvajalci obvezali in striktno sledili tem priporočilom.

3.1.3 Avtomobili

Vsak avtomobil ima v sebi računalnik, ki nadzira njegovo delovanje. In kjer so računalniki, najverjetneje obstajajo tudi varnostne pomanjkljivosti. Za preizkus so se že leta 2010 odločili raziskovalci University of Washington and University of California-San Diego, ki so odkrili da lahko precej enostavno vdrejo v računalniški sistem avtomobila in z njim upravljajo več ali manj vse funkcije. [23] Uspeli so onemogočiti zavore, zakleniti vrata, nadzirali so radio, klimatsko napravo in med drugim tudi hupo.

Za vdor v računalniški sistem so sicer potrebovali fizičen dostop do avtomobila. Uporabili so On-Board Diagnostics (OBD-II) priključek, ki ga ima vsak avtomobil. To jim je omogočilo dostop do Controll Area Network vodila (CAN bus). CAN je protokol ki omogoča mikrokontrolerjem in ostalim napravam komunicirati med sabo brez glavnega računalnika. Ko so imeli dostop do CAN vodila, so lahko upravljali z različnimi Elektronskimi nadzornimi enotami (Electronic Controll Units – ECU). V sistem je bilo sicer vgrajeno nekaj varnostnih elementov, a so jih raziskovalci obšli brez večjih težav.

Današnji avtomobili imajo lahko do preko 70 takih enot [24], namenjene pa so kontroliranju električnih sistemov ali podsistemov v vozilu. Največja in najpomembnejša nadzira delovanje motorja, ostale pa so tudi za nadzor hitrosti, varnostnih vreč, zavor, eklektično premičnih stekel, vrat, ogledal… Nekatere so sicer neodvisne med sabo, ampak je komunikacija med njimi vseeno potrebna. Nekatere pa komunicirajo med sabo preko CAN protokola in taka komunikacija nam omogoči, da se zbere več podatkov o delovanju vozila (hitrost, kot volana,

klimatska naprava…) in se glede na te podatke naprimer izključi delovanje motorja ko avto stoji pri miru. S tem avto varčuje na gorivu.

Raziskovalci so ugotovili da lahko na nekatere ECU sisteme namestijo novo strojno programsko opremo tudi med samim delovanjem avtomobila. S tem pa povzročijo ponovni zagon motorja, kar je lahko že nevarno. Raziskovalcem je uspelo med vožnjo vključiti brisalce, hupati brez prestanka, vklapljati in izklapljati radio, upravljati s klimatsko napravo pa tudi upravljati z zavorami vozila. Testirali so in izvedli test, kjer so pri premikajočem avtomobilu vklopili brisalce, a šele ko je dosegel določeno hitrost. Niso pa poskušali nevarnih stvar, kot je popolna izključitev zavor. Kar bi tudi bilo mogoče z njihovim dostopom. Še več, povedali so, da bi lahko zagnali ponovni zagon ter obnovili vso njihovo strojno opremo nazaj na prvotno in tako bi vozilo pustili brez sledi. Edina stvar ki je pa niso mogli spremeniti je upravljanje z volanom vozila. Potrdili pa so, da je to mogoče če ima avtomobil možnost samodejnega parkiranja.

Slika 9: Jeep Cherokee

Podobna, a še hujša zgodba se je pripetila tudi avtomobilskemu proizvajalcu Jeep z modelom Cherookee. [25] Dva raziskovalca, Charlie Miller in Chris Valasek, sta uspela na daljavo prevzeti popoln nadzor nad vozilom. Na daljavo lahko sta lahko vsakemu modelu tega vozila izključila motor, se igrala z radiem, klimatsko napravo, zavorami in ostalim. Njihov program pa ne omogoča samo prevezama nadzora nad vozilom ampak lahko brez vednosti voznika vozilo tudi sledijo, beležijo hitrost in pozicijo. Vse to je možno, ker je velika večina avtomobilov povezanih v internet. Napada pa je tudi zanimiv, ker jima je omogočil

upravljanje z katerokoli vozilom modela Jeep Cherookee (Slika 9), proizvedenim med letim 2013 do 2015, ki uporabljajo sistem Uconnect. Dejala sta, da če vemo IP naslov avtomobila, lahko napad lociramo na točno določeno vozilo. Demonstrirala sta tudi skeniranje za temi vozili in jih nekaj tudi odkrila. To so jih vozili ljudje, ki niso vedeli da jih kdo spremlja.

Da bi težavo čimprej rešili so obvestili proizvajalca avtomobilov Chrysler, ki je vpoklical 1,4 milijona vozil, ki so bila v nevarnosti, da odpravi težavo. Problem pa je, da težava ni rešljiva z avtomatsko posodobitvijo na daljavo, ampak je potreben fizičen dostop do avtomobila. To pomeni, da se kljub prizadevanju prodajalca, da se napravo čimprej odpravi, še vedno po cestah vozi veliko nezaščitenih avtomobilov. To pa je problem, ki ga je najlaže reševati kot pri zdravstvenih sistemih – z nadgradnjo sistema na daljavo, saj bi tako lahko zagotovili da so vse odkrite varnostne luknje zakrpane takoj, ko se zanje najde rešitev in nebi bilo potrebno avtomobilov voziti na servise, kjer bi morali vsak avtomobil priklopili na računalnik in jih posodobiti ročno.

3.1.4 Elektronska varuška

Raziskovalna ekipa z imenom Rapid7 so se odločili da testirajo 9 modelov elektronskih varušk, ki so povezane v internet in v vseh devetih testiranih primerkih so odkrili varnostne pomanjkljivosti. [26] Nekaj varnostnih pomanjkljivost je navedenih spodaj.

Problem če nekdo nepooblaščeno dostopa do elektronske varuške ni tako velik kot če nekdo nepooblaščeno dostopa do avtomobila (kar smo videli v prejšnjem poglavju), a vendar stvar ni tako nedolžna. Varnostne luknje v teh napravah so omogočile da ljudje iz drugega konca sveta preko interneta spremljali video sliko v živo, poslušali zvok in ga tudi predvajali. Omogočeno jim je bilo spreminjanje nastavitev kamere in dodajanje pooblaščenih oseb ki lahko dostopajo do video nadzora otroka.

Raziskovalci ekipe Rapid7 so opozorili na to, da nepooblaščen dostop do podatkov iz elektronske varuške ni tako nedolžen kot se zdi na prvi pogled. Mogoče nas ne moti toliko, če kdo sliši kaj se pogovarjamo, a problem nastane če na primer od doma dela direktor kakšnega podjetja, takrat pa mu lahko napadalec prisluškuje in izve marsikatero dejstvo, ki bi ga lahko uporabil proti njemu in njegovemu podjetju. To pomeni, da lahko varnostna luknja v tako na videz nedolžni napravi kot je elektronska varuška pripomore do odkritja varnostne luknje v večjem sistemu nekega podjetja.

Elektronske varuške so testirali za najbolj znane ranljivosti, kot so privzeti uporabniški račun in geslo, nešifrirano pošiljanje video ter avdio podatkov ter razni ukazi poslani v čistopisu.

Odkrili so več problemov elektronskih varušk, s katerimi lahko dobimo dostop do njih ali pa podatkov o videu:

- Direktna povezava na strežniški del elektronske varuške, ki je nešifrirana ter ne zahteva avtentikacije uporabnika. Nepooblaščena oseba lahko z napadom z grobo silo locira kamero, kar ji omogoča gledanje video prenosa kamere v živo pa tudi spreminjanje nastavitev kamere. To naredi tako, da na omrežju preverja vse kombinacije imena gostitelja ter številko vrat preko katerih elektronska varuška komunicira s ponudnikom storitve. To ranljivost so odkrili pri elektronski varuški proizvajalca Phillips, ki pa se je odzval na njihov kontakt o odkriti ranljivosti in je ranljivost tudi odpravil s tem, da je šifriral komunikacijo naprave s ponudnikom storitve.

- En model varuške proizvajalca iBaby je imel spletni portal, kjer pa so ugotovili, da lahko z enostavnim spreminjanjem serijske številke naprave v URL nizu dostopajo do podatkov naprav vseh ostalih elektronskih varušk tega modela. Napadalci bi lahko s pomočjo te varnostne pomanjkljivosti in programa, ki bi zaporedno preverjal vse URL naslove pridobili nepooblaščen dostop do vseh videoposnetkov teh elektronskih varušk. Jaz menim, da bi bila rešitev za tak problem to, da mora spletni strežnik avtenticirati vsak zahtevek in določiti, ali mu prikaže vsebino, ali mu zahtevo zavrne. Tak način je tudi prikazan v moji aplikaciji v zadnjem poglavju.

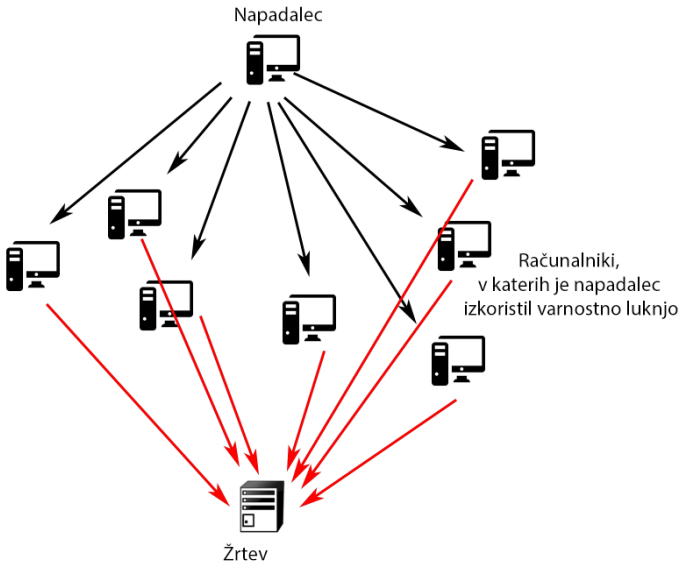
- Spletni del elektronskih varušk proizvajalca Summer Infant je vseboval ranljivost, kjer dodajanje dovoljenja novemu uporabniku za dostop do kamere ni zahtevalo avtentikacije. To je pomenilo, da si lahko dodal dovoljenje le z klicem določene ga URL naslova. poznati si moral le identifikacijsko številko uporabnika, ki že ima to pravico dodajanja ter nov elektronski naslov, ki mu bo ta pravica dodeljena. Tako je lahko napadalec s pomočjo skripte in iteracije URL naslovov dodal nov elektronski naslov vsem elektronskim varuškam tega modela in si omogočil gledanje video posnetkov v živo na vseh.

Problem elektronskih varušk pa tudi ostalih naprav ki spadajo v internet stvari je da povečini tečejo na raznoraznih verzijah Linux sistema. Ta pa je zastarel kmalu po izidu na trg, posodablja pa se ga ne več. Kar pomeni, da je na svetu ogromno starejših naprav povezanih v internet ki imajo ranljivosti za katere danes vemo in jih poznamo. Kot zanimivost so sporočili da se je samo eden proizvajalec elektronskih varušk odločil da jih posodobi, ostali pa se niso odzvali na njihov kontakt ali pa so bili nedosegljivi. Zaradi tega so priporočali da se kupuje elektronsko varuško, ki ni povezana v internet, če to ni potrebno. Sicer so tudi take elektronske varuške ranljive, saj se podatki prenašajo brezžično med varuško in prenosno

napravo, a vendar tukaj mora biti napadalec v dometu naprave, kar pa ni potrebno, če je elektronska varuška poveza v internet.

3.2 DDoS napadi s pomočjo interneta stvari

V prejšnjem poglavju sem opisal štiri primere napadov na internet stvari, a ti napadi so izkoristili varnostne luknje tako, da so napadalci pridobili dostop do točno določene naprave. Napadalci so pa te varnostne luknje izkoristili tako, da so imeli možnost oškodovati samo osebe ki so uporabljale napravo (ali bile v njeni bližini). Če v prejšnjih primerih nekdo ni uporabljal naprave, pri kateri je bila odkrita varnostna luknja, potem nanj ta varnostna pomanjkljivost ni vplivala. Spodaj pa si bomo pogledali še ene vrste napada, ki pa lahko vpliva tudi na osebe, ki ne uporabljalo ranljivih naprav.



Slika 10: Diagram napada s porazdeljeno zavrnitvijo storitve

DDoS (Distributed Denial of Service) ali po slovensko napad s porazdeljeno zavrnitvijo storitve je napad, kjer je cilj onemogočiti delovanje enega računalnika ali računalniškega sistema. Kot prikazuje Slika 10. Napadalec najprej pod svojo kontrolo prevzame mnogo drugih računalniških sistemov, ti pa potem pošiljajo veliko zahtevkov na en določen strežnik. Ti zahtevki potem tako povečajo promet do napadenega strežnika, da napaden strežnik ne zmore več obdelati vseh. Zaradi tega se zgodi, da so tudi legitimni zahtevki pravih uporabnikov zavrnjeni. Uporabnikom se torej zdi da je storitev počasna ali pa da ne deluje.

Za napade s porazdeljeno zavrnitvijo storitve pa se lahko uporablja tudi naprave iz interneta stvari, ne samo osebne računalnike. Zlorabljene naprave potem lahko zajemajo vse od spletnih kamer, televizorjev, hladilnikov in raznoraznih senzorjev, ki so priključeni v internet. Napadalec lahko dobi dostop do naprave s preučevanjem varnostnih pomanjkljivosti in ker je naprav priklopljenih na internet veliko, je veliko lukenj tudi znanih. Še ena opcija pa je, da se poskuša prijavit v nadzorno ploščo naprave z privzetim uporabniškim imenom in geslom. Ko ima nadzor nad napravami pa lahko z njihovo pomočjo sproži DDOS napad, tak napad pa je bil tudi t.i. Mirai, ki je opisan v nadaljevanju.

Mirai [27] je zlonamerna programska oprema, ki je računalniške sisteme, ki tečejo na operacijskem sistemu Linux spremenila v botnet omrežje – omrežje povezanih naprav, nad katerimi napadalec prevzame nadzor, da jih lahko uporabi za izvajanje zlonamernih dejanj kot je DDOS napad. Velika večina naprav, ki jih je zlonamerna koda Mirai okužila, so domače varnostne kamere ter usmerjevalniki.

Mirai deluje tako, da vsaka okužena naprava skenirala omrežje za drugimi napravami v internetu stvari in ko je tako napravo našla, se je poskusila prijaviti vanjo s privzetim uporabniškim imenom in geslom. Ko ji je to uspelo, je napravo okužila, a naprava je vseeno delovala enako kot prej – mogoče z izjemo občasnega slabšega delovanja ter večje porabe pasovne širine, saj je okužena naprava v splet pošiljala spletne zahtevke in tako sodelovala v DDOS napadu.

Mirai je bil uporabljen v enem večjih DDOS napadov 20 Septembra 2016. Do takrat je bil to eden večjih DDOS napadov nasploh, saj je dosegel do 620Gb na sekundo. Dober mesec kasneje so tudi kodo Mirai uporabilia za velik napad na Dyn. To je podjetje, ki nadzira dobršen del infrastrukture sistema domenskih imen (DNS). Napad nanj je za določen čas onesposobil kar nekaj večjih spletnih portalov, med drugim tudi Twitter, Netflix ter GitHub. Kasneje leta 2016 pa je Mirai povzročil sesutje skoraj milijon usmerjevalnikov s tem, ko je poskušal izkoristiti varnostno luknjo. S tem je povzročil nedelovanje usmerjevalnikov proizvajalca Zyxel ter Speedport, proizvajalec pa je zakrpal težavo in uporabnikom priporočil da naj usmerjevalnik izključijo iz napajanja za 30 sekund, nato pa naj ga ponovno vključijo, da se posodobi strojna programska oprema. [28]

4. Varnost in zaščita

V prejšnjem poglavju smo videli nekaj primerov napadov na internet stvari, v tem poglavju pa bo opisano zakaj je težje zagotoviti varnost napravam interneta stvari kot na primer mobilnim telefonom ali osebnim računalnikom.

V devetdesetih letih prejšnjega stoletja so osebni računalniki pridobivali na popularnosti. Tudi takrat pa je programska oprema vsebovala varnostne ranljivosti. [29] Ko so proizvajalci programske opreme našli varnostno ranljivost in jo zakrpali, je velik izziv predstavljal kako tak varnostni popravek namestiti na osebni računalnik nekega uporabnika. Danes je vse skupaj malo lažje (vsaj za pametne mobilne telefone ter osebne računalnike), saj ima dostop do interneta že slaba polovica svetovne populacije, [30] operacijski sistemi in aplikacije pa imajo implementirane samodejne posodobitve. Kar pomeni, da ko proizvajalec programske opreme najde napako in jo odpravi, se ta posodobitev v zelo kratkem času prenese do uporabnikov. Lep primer so današnji operacijski sistemi kot so Android, iOS ali pa Windows, ki se posodabljajo avtomatsko, brez da bi uporabnik moral za posodobitve zagnati poseben postopek. [31]

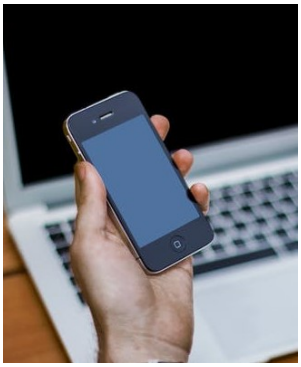
Kljub temu da smo v prejšnjem odstavku videli, da za nekatero programsko opremo obstajajo avtomatske posodobitve, pa pri nekaterih napravah, ki so povezane v internet le te še ne obstajajo (navedene v poglavju 3). In če vzamemo v obzir, da so računalniki, v na primer usmerjevalnikih zmogljivejši od računalnikov pred dvajsetimi leti, vidimo, da nam take naprave predstavljajo veliko varnostno tveganje. Moje osebno opažanje je, da se ljudje iz moje okolice, torej študenti ter starejši, ki jim računalništvo ni blizu, ne zavedajo, da so ranljivi tudi na primer usmerjevalniki, ne samo osebni računalniki. Taki zato menijo, da v usmerjevalnike ni mogoče vdreti, kar pa je ravno nasprotno dokazal raziskovalec Craig Heffner, ki je na varnostni konferenci Def Con predstavil rezultate svoje raziskave. Testiral je 30 usmerjevalnikov, vdreti pa mu je uspelo v polovico njih. Med drugimi so bile to tudi poznani proizvajalci kot so Linksys, Dell, Belkin in še nekateri drugi. [32] Za napad je uporabil DNS vnovično povezovanje [33].

Slika 11: Primer usmerjevalnika

4.1 Zakaj naprave ostanejo nezaščitene?

Veliko naprav interneta stvari vsebuje posebej dizajnirane računalniške čipe, ki jih proizvaja nekaj podjetij (Broadcom, Qualcomm…) [34]. Te čipe pri izdelavi naprave uporabljajo proizvajalci strojne opreme, katerih ime pa ni na izdelku, ampak napravo samo izdelajo ter dajo naprej tistemu podjetju, pod katerim imenom se bo naprava prodajala. [35] Naprava se proda in dokler je podpora se tudi krpajo varnostne luknje. Naprave, ki jih uporabljamo potrošniki, pa proizvajalci s časom nehajo podpirati.

Za primer si lahko pogledamo mobilni telefon iPhone 4 proizvajalca Apple. [36] Mobilni telefon je izšel leta 2010, zadnja posodobitev operacijskega sistem pa je prejel leta 2013. Zadnjo posodobitev so uporabniki prejeli 4 leta nazaj in kdor še vedno uporablja ta telefon je izpostavljen varnostnim tveganjem. V operacijskem sistemu 7.1.2, ki ga uporablja iPhone 4 so bile odkrite varnostne luknje in tudi javno objavljene, ampak Apple do sedaj ni izdal posodobitve za ta sistem. [37]



Slika 12: iPhone 4

Tako kot pri mobilnem telefonu Apple iPhone 4, ki ga prikazuje Slika 12, lahko po mojem mnenju posplošimo to trditev na še katero napravo interneta stvari (tudi na Jeep Cherookie vozilo, kot smo videli v poglavju 3.1.3).

4.2 Varnost z načrtovanjem v naprej

Ena izmed možnih rešitev, kako lahko za naprej poskrbimo za varnost interneta stvari je izdelava naprave z mislijo na varnost že od samega začetka razvoja, torej od ideje dalje. [38] Naslednji korak je tudi ta, da tak proces vpeljave varnosti v sistem določeno neodvisno podjetje tudi preveri in izda certifikat. Tako se lahko zagotovi, da je vse narejeno skladno s pravili.

V grobem bi moral razvoj vsake naprave vključevati vsaj tri faze:

- **Faza analize:** V tej fazi proizvajalec razišče kako bodo napravo uporabljali končni uporabniki. Na podlagi teh določil tudi že lahko določi nekatere varnostne probleme, ki jih bo moral rešiti. Za primer je v raziskavi uporabljen naveden, ne pameten termostat. Tak termostat skrbi samo za vzdrževanje željene temperature v stanovanju in ne predstavlja nobene varnostnega tveganja. Za kakršnokoli spremembo temperature je potrebno fizično, na termostatu, spremeniti željeno temperaturo. Pameten termostat, kot je na primer NEST in ga prikazuje Slika 13, ki pa je povezan v internet in se lahko preko interneta tudi regulira temperaturo in upravlja z vsemi njegovimi funkcijami, pa je že potreben zaščite. Za to je potrebno da se določi, da lahko s termostatom (ali katerokoli napravo) upravlja le pooblaščena oseba, ki ima dovoljšne pravice.

Slika 13: Pameten termostat podjetja NEST, ki je že bil tarča napada.

- **Faza planiranja:** V tej fazi proizvajalec določi vse možne načine, kako lahko uporabnik upravlja z napravo. Če vzamemo primer iz prejšnje faze, torej s termostatom. V primeru termostata ki ni pameten in ne povezan v internet, uporabnik spreminja temperaturo in ostale funkcije le tako, da fizično na termostatu pritiska gumbe. Na pametnem termostatu pa je situacija malo drugačna, saj lahko temperaturo in ostale nastavitve upravlja iz več sistemov in aplikacij.
    - Temperaturo in ostale nastavitve lahko spreminja ročno, kot pri termostatu ki ni pameten
    - Lahko uporablja mobilno aplikacijo, ki je preko lokalnega omrežja povezana z termostatom in preko nje upravlja z njim vse njegove nastavitve.
    - Če je termostat povezan v internet, lahko uporabnik spreminja temperaturo in ostale nastavitve preko spletne aplikacije, vsi ti podatke pa gredo preko strežnika proizvajalca.
- **Faza implementacije:** V tej zadnji fazi proizvajalec pregleda vse mogoče varnostne luknje, ki bi se lahko pojavile – te je ugotovil v prejšnjih dveh fazah in jih prepreči. Nekaj pogostejših načinov, ki jih mora proizvajalec pri tem opraviti je, da:
    - Zagotovi, da strežnik prepozna in pravilno identificira uporabnika naprave. Največkrat to preko uporabniškega imena in gesla.
    - Na napravi preveri, če zahtevek prihaja iz pravilnega strežnika. S tem se zaščiti da lahko ukaze na napravo pošiljajo samo določeni strežniki, ne pa strežniki napadalcev.

- Preveri identiteto aplikacije, preko katere se bo upravljalo z napravo ter preveri, če ima aplikacija sploh dovolj velike pravice.
- Zagotovi, da je spletni strežnik prepričan, da so pravice, ki jih ima naprava in so shranjene v strežniku proizvajalca prišle iz pravega termostata.
- Da noben drug, razen dovoljenih uporabnikov nima dostopa do naprave.

Celoten postopek sicer ne zagotavlja da je naprava varna, a vendar precej minimizira možnost napada nanjo, saj se veliko varnostnih pomanjkljivosti ugotovi že vmes.

4.3 Organizacija OWASP

Kako bi poskrbeli za varnost pri internetu stvari, z upoštevanjem metode »Security by design« oz. varnosti z načrtovanjem v naprej, pa so objavili principe tudi pri organizaciji OWASP [39] (Open Web Application Security Project). OWASP (Slika 14) je neprofitna organizacija, ki piše prosto dostopne članke, metodologije, dokumentacije in razna orodja na področju spletne varnostni, njen cilj pa je izobraževati razvijalce ter ostale sodelujoče pri projektih o najbolj pogostih varnostnih tveganjih na področji interneta.

Slika 14: Logotip OWASP organizacije.

Organizacija je klasificirala najpogostejše vrste napadalcev na sisteme od najbolj verjetnih in pogostih do najmanj. In ti si sledijo:

- Nezadovoljno osebje in razvijalci
- Stranski učinki in posledice trojanskih konjev, črvov in ostalih virusov.
- Organizirani napadalci na računalniške sisteme z motivacijo napada na točno določen sistem
- Napadalci brez neke točne motivacije
- Napadalci, ki se še učijo napadati.

Poleg tega so tudi objavili 3 glavne stebre na katere moramo paziti pri varnosti informacij. [40] Ti trije pomembni pa so:

- **Zaupnost:** Pomeni, da informacije niso na voljo osebam, ki sicer ne bi imeli dostopa do teh informacij. En primer kako zagotovimo zaupnost je šifriranje podatkov, za dostop pa se uporabnik na primer prijavi z uporabniškim imenom in geslom. Vedno več se sicer uporablja dvostopenjska avtentikacija, ostali načini pa so še biometrični senzorji (zenica, prstni odtis), prijava s certifikatom…

- **Integriteta:** Pomeni, da podatki pridejo nespremenjeni od pošiljatelja do prejemnika. Zagotoviti moramo, da nepooblaščena oseba nima možnosti spreminjati podatkov med prenosom. Take probleme lahko rešimo s pravicami. Med prenosom pa se lahko podatki spremenijo tudi zaradi motenj v strojni opremi, zato se uporabljajo razne metode za zaznavo spremembe podatkov.

- **Dostopnost:** Zagotovimo, da so sistem in podatki na voljo avtoriziranemu uporabniku, ko jih le ta potrebuje. Dostopnost najlažje zagotovimo tako, da stalno vzdržujemo strojno opremo, da popravimo in menjamo dele takoj, ko je potrebno ter vzdržujemo in stalno posodabljamo programsko opremo ki jo ima naprava nameščeno. Ker se nezgoda lahko vseeno zgodi je pametno imeti redundantne podatke na več fizično ločenih lokacijah in pa dovolj hitro povezavo, da uspemo v doglednem času servirati vse zahtevke.

4.3.1 Varnostni principi

OWASP organizacija je za vsako napravo v internetu stvari priporočila, da se jo dizajnira ter proizvede s temi principi. Tako bodo naprave veliko bolj varne kot če teh principov proizvajalec nebi upošteval:

- **Minimizirajmo točke napada:** Vsaka dodatna funkcija, ki jo omogoča program predstavlja novo tveganje za vdor v sistem. Zato je cilj minimizirati možnosti za napad v dodatnih funkcijah programa. Primer ki ga navajajo je, da lahko imamo spletno stran, ki ima omogočeno iskanje. Tako iskanje je lahko ranljiva za napade s vrivanjem SQL stavkov. Če iskanje omejimo le na avtorizirane uporabnike, potem je verjetnost da pride do takega napada precej manjša. Če iskalni niz pred izvedbo še preverimo, ali zadostuje določenim pogojem, potem možnosti za napad zelo zmanjšamo. V najboljšem primeru pa, če imamo to možnost, že sam uporabniški vmesnik spletne strani zasnujemo tako, da kar rabimo iščemo drugače, ne z vnosnim poljem.

- **Varne privzete nastavitve:** Naprava naj ima vključene čimbolj varne privzete nastavitve, nato pa naj dovoli uporabniku (če mu dovoli), da te nastavitve spreminja in jim zmanjšuje varnost. Na primer, naprava naj na začetku zahteva čimbolj varno geslo. Če pa uporabnik izrecno želi, pa lahko kakšen pogoj za geslo odstrani in si lahko nastavi manj varno geslo – a uporabnika je treba na to tudi opozoriti. Seveda to ne pomeni da se naprava privzeto ne povezuje nikamor in ima vse povezave blokirane, ampak vsaj toliko, da dela kot mora.

- **Čim manj pravic:** Ta princip priporoča, da ima račun, ki izvaja neke akcije le minimalno potrebno število pravic za uspešno izvedbo. V to se šteje uporabniške pravice kot tudi pravice sistema za dostop do strojnih virov (procesor, spomin, omrežje) in pravic do datotek. Na primeru, če nek vmesni člen za svoje delovanje potrebuje samo dostop do omrežja ter pisanje v podatkovno bazo, sta le ti dve pravici ki mu jo dodelimo in nič več.

- **Več stopenj obrambe:** Ta princip nam določa, da imamo v sistemu implementiranih več stopenj zaščite. Tako tudi če napadalec povzroči odpoved ene od naših zaščit, imamo še več drugih ki še vedno napadalcu preprečujejo vstop v sistem. Kot primer je navedeno, da se ni pametno zanašati na požarni zid, da bo dovolj za varnost aplikacij ki se uporabljajo znotraj lokalnega omrežja. Požarne zidove lahko izkušen napadalec zaobide s socialnim inženiringom ali pa z grobo silo, če ima dostop do strežnika. V tem primeru je dobro imeti tudi fizično varnost strežnikov in nadzorne kamere ter usposabljanje zaposlenih.

  Včasih z dodajanjem nove stopnje varnostni v sistem ne pridobimo dosti, a zelo povečamo kompleksnost aplikacije. Zaradi tega moramo vedno pretehtati če bo željena dodatna varnostna zaščita res toliko povečala varnost kot si želimo. Če na primer sistem zahteva 15 črk dolgo geslo in še več različnih znakov, namesto prej le 8 črk dolgo, si ga bodo uporabniki mogoče zapisali na list papirja in s tem ne povečamo, ampak celo močno zmanjšamo varnost. Če pa dodamo vpis preko pametne kartice, pa se s tem varnost celotnega sistem precej poveča.

- **Varno lovljenje napak:** Napake in izjeme se dogajajo in zgodijo se lahko tudi če ni naša napaka in nimamo vpliva nad tistim delom. Recimo če želimo kaj zapisati v podatkovno bazo, a se baza ne odziva. Zaradi tega je pomembno da na vseh delih kode, kjer se napake lahko zgodijo pametno upravljamo z izjemami. Ko implementiramo varnostne mehanizme imamo po večini na voljo tri različne rezultate:

  - Dovolimo operacijo

- Ne dovolimo operacije
- Izjema

V splošnem stremimo k temu, da se izvajanje programa po napaki izvede kot da nam pravice za željeno operacijo niso bile dodeljene. V primeru napake naj metoda, ki na primer preverja pravilnost vnosa vrne »false«, če pa vrne napako, pa moramo točno vedeti zakaj je bila vrnjena ta napaka in kako ukrepati naprej. Paziti pa moramo tudi da neka povzročena napaka v programski kodi ne povzroči tega, da se kasnejša programska koda ne izvede. To je pokazano na primeru spodaj:

```
isAdmin = true;
try {
  codeKiVrneNapako();
  isAdmin = isUserInRole( "Administrator" );
}
catch (Exception ex)
{
  log.write(ex.toString());
}
```

V zgornjem primeru, ko se izvede metoda ki vrne napako, se naslednji korak preskoči in izvajanje programa se nadaljuje v »catch« delu. To pomeni, da ima aplikacija še vedno pravice administratorja, kar pa ni vredu. Precej bolje bi bilo, če bi na začetku nastavili da uporabnik ni administrator, in če bi mogoče metoda potrebovala administratorske pravice, bi lahko pred klicem preverili če je uporabnik administrator.

- **Ne zaupati zunanjim sistemom:** Veliko sistemov, aplikacij in organizacij uporablja izkorišča ponudbo zunanjih partnerjev za procesiranje ali delo z podatki. Taki partnerji pa imajo najverjetneje drugačno varnostno politiko od lastnega podjetja, zato moramo biti pri podatkih ki jih vračajo izredno previdni. Če moramo določen podatek prikazati uporabniku, moramo tudi mi samo preveriti ali je podatek pravi, a ni prevelik, mogoče premajhen od pričakovanega.

- **Ločevanje dolžnosti:** Ta princip nam na grobem predlaga, da naj niso tisti, ki potrdijo neko akcijo, tisti ki jo izvajajo ter tisti ki jo nadzirajo, nikoli ista oseba. To izloči, da bi ena oseba izvedla akcijo, ki jo sicer nebi smela. Na primer, če ena oseba naroči nov računalnik in je tudi podpisnik, da je računalnik bil dostavljen se lahko zagovarja, da računalnik ni nikoli prišel. Če pa sta zraven še dve drugi osebi, ki to nadzirata, pa to ni več mogoče. Prav tako na primeru računalniških sistemov, kjer ima administrator pravice za izključiti ter zagnati sistem, spreminjati nastavitve, ne sme pa imeti možnosti se prijaviti kot drug uporabnik in na njegov račun kupovati stvari.

- **Zagotavljanje varnost s skrivanjem:** Zelo šibek način varnosti je ta, da skrivamo način implementacije sistema ali programa. Če se zanašamo na to, je velika verjetnost da bo način prej ali slej odkrit in zaradi tega se samo na to ne smemo zanašati. Primer je Linux sistem, kjer je vsa programska koda znana a je še vedno ob pravilnih implementacijah zelo varen in robusten.

- **Čimbolj enostavno:** Če imamo možnost izbire med zelo kompleksnim sistemom, ki doda malo na varnost ter zelo enostavno a še vedno zelo varno rešitvijo, je precej bolje implementirati enostavnejšo rešitev. Sistem bo tako hitrejši in manj verjetno je da smo v njega vpeljali nove varnostne luknje.

- **Pravilno krpanje lukenj:** Ko je enkrat varnostna luknja znana jo je potrebno čimprej zakrpati. Potrebno je vedeti točen vzrok za njen nastanek ter napisati nekaj testov in preveriti če mogoče popravek ne povzroča novih lukenj. Kot primer, če neko napako povzroči spremenjena vrednost piškotkov in je popravek lahko enostaven, se moramo zavedati da lahko ta piškotek uporablja še veliko drugih sistemov in moramo paziti, da jih ne pokvarimo.

5. Primer prikaza varnega razvoja na konkretni aplikaciji

Namen praktičnega dela diplomske naloge je razviti delujočo mobilno aplikacijo, s katero bom lahko ocenil količino dela, ki ga zahteva vzpostavitev varne povezave med aplikacijo ter spletnim strežnikom. Med drugim pa je tudi cilj preveriti, katere vrste napadov aplikacija razvita na tak način preprečuje in na katere je še ranljiva. Iz dejstev, ki jih bom spoznal pri razvoju aplikacije, menim, da bom lahko ocenil, do katere stopnje je še potrebno zagotavljati varnost, če razvijamo aplikacijo, ki na primer beleži prekolesarjeno pot uporabnika in te podatke pošilja v spletni strežnik.

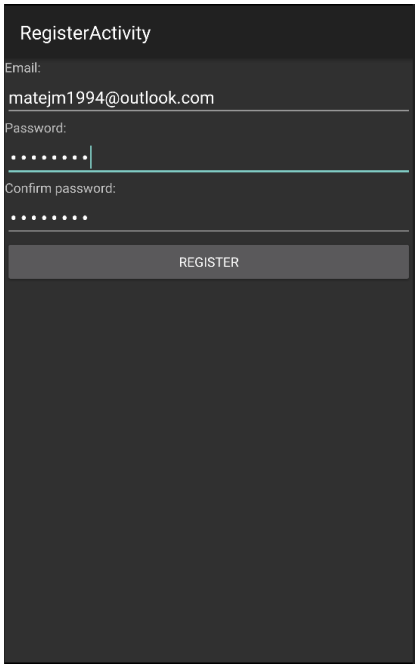
Za ta namen bom razvil mobilno aplikacijo, ki bo z uporabo GPS beležila pozicijo uporabnika ter te podatke prenašala v spletni strežnik, kjer jih bo tudi shranjevala. Mobilna aplikacija bo izdelana v Xamarainu v C# jeziku, spletni strežnik v .net WebApi okolju, gostuje pa na lokalnem IIS. Zakaj sem izbral točno določena orodja in programske jezike je opisano v nadaljevanju.

Dva možna napada na aplikacijo bi bila napad z grobo silo, kjer bi napadalec z ugibanjem preverjal pravilnost gesla, ali pa zajemanje prometa med aplikacijo ter spletnim strežnikom, v primeru ne šifrirane povezave med njima. Verjetnost za obe vrsti napada na aplikacijo sem zmanjšal s tem, da sem med aplikacijo ter spletnim strežnikom vzpostavil šifrirano HTTPS povezavo, pri registraciji uporabnika pa je zahtevano geslo, dolgo vsaj 8 znakov, z vsaj eno veliko, vsaj eno malo črko ter poseben znak in številko.

5.1 Mobilna aplikacija

Cilj mobilne aplikacije je prikaz povezave med mobilnim telefonom s spletnim strežnikom ter prenos podatkov med njima. Uporabnik se mora pred uporabo aplikacije registrirati, nato pa se lahko prijavi v sistem ter beleži svoje podatke o treningu. Ko konča lahko podatke zavrže ali pa jih prenese v spletni strežnik. Vse podatke, ki jih shrani v spletni strežnik, jih ima vedno na voljo tudi za pregled. Spletni strežnik pa mu naredi tudi malo bolj podrobno analizo podatkov.

Ko uporabnik zažene aplikacijo, ima najprej na voljo dva gumba. Enega za prijavo, drugega za registracijo. Če uporabnik še ni prijavljen, se mora najprej registrirati. To stori tako, da izbere registracijo in odpre se mu novo okno. Tam vpiše svoj elektronski naslov, geslo ter geslo tudi potrdi. To prikazuje Slika 15.

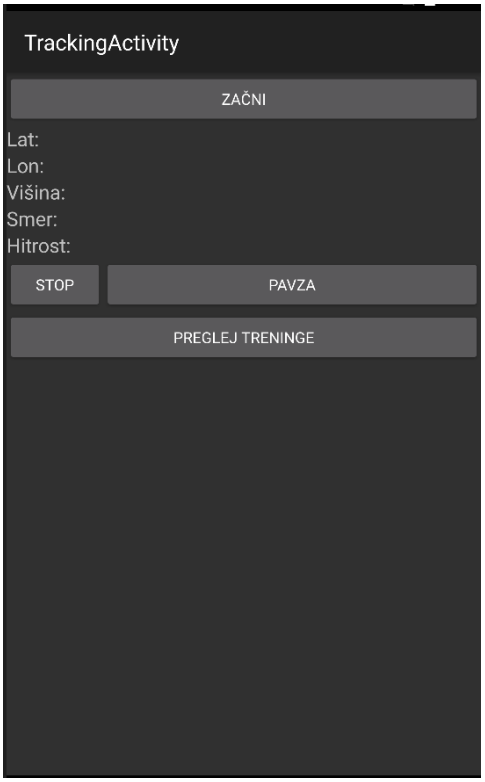


Slika 15: Registracija

Ko uporabnik klikne gumb za registracijo, se na spletni strežnik pošlje zahteva preko *https* povezave. Spletni strežnik preveri, če elektronski naslov še ne obstaja, če se gesli ujemata ter zadostujeta pogojem. Če je vse v redu, je registracija uspešna in uporabnika aplikacija preusmeri na prijavno stran. Tam vpiše svoje podatke in lahko se v aplikacijo prijavi.

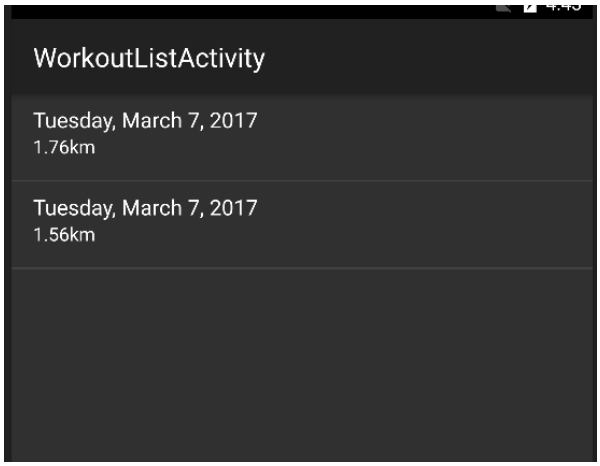
Ko se uporabnik uspešno prijavi se mu prikaže glavna stran, kot jo prikazuje Slika 16. Na vrhu ima uporabnik gumb »Začni«, ki začne beleženje njegove pozicije in podatkov o hitrosti, nadmorski višini ter smeri gibanja, če mobilni telefon to podpira. Smer gibanja pa je številka med 0 in 360, ki predstavlja kot med severom in določeno točko. Ko se uporabnik premika, se to premikanje tudi beleži in shranjuje v mobilnem telefonu.

Med treningom se lahko uporabnik odloči ustaviti aplikacijo, ki tako ne beleži več premikanja, a podatki še vedno ostanejo shranjeni v telefonu, tako da lahko nadaljuje dokler ima aplikacijo odprto. Lahko pa se odloči trening zaključiti, kar pa naredi s pristiskom na gumb *STOP*. Ko pritisne na gumb se odpre dialog, ki ga vpraša če želi podatke shraniti ali izbrisati. Če jih shrani, se podatki prenesejo v spletni strežnik (imeti mora internetno povezavo), če pa jih zavrže pa se podatko izbrišejo.



Slika 16: Glavna stran programa

Naslednja funkcija programa pa je gumb *»Preglej treninge«* , ki odpre seznam vseh do sedaj opravljenih treningov. (prikazano na Slika 17)



Slika 17: Prikaz treningov

Na sliki vidimo da sta opravljena dva treninga v dolžini ki je podana spodaj, pod datumom, ki je hkrati tudi ime treninga. Ti podatki o treningih se prenesejo iz spletnega strežnika, zato jih

ima uporabnik vedno pri sebi, tudi če odstrani aplikacijo in jo na novo namesti ali pa na primer zamenja mobilni telefon.

Zadnja funkcionalnost android aplikacije pa je malo bolj podroben vpogled v podatke treninga. Če uporabnik klikne na katerokoli trening, pa se mu odpre novo okno, ki mu prikazuje podrobne podatke o treningu. To okno prikazuje Slika 18, ker pa sem v tem primeru podatke beležil z uporabo posnemovalnika, je nadmorska višina 0, saj je posnemovalnik ne podpira.

Slika 18: Podrobni opis

Vsi prikazani podatki se prenesejo iz spletnega strežnika. Tam se tudi izračunajo vsi podatki, kot so na primer povprečna hitrost ali maksimalna nadmorska višina. S tem mobilni telefon razbremenimo dela in prihranimo na bateriji, kar je pri večji količini podatkov zelo pomembno.

5.2 Spletni strežnik

Spletni strežnik je narejen na tehnologiji *ASP.NET WebApi,* na podlagi predloge, ki za avtentikacijo uporabnikov uporablja individualne uporabniške račune. To predlogo sem izbral, saj mi je olajšala prijavo in registracijo uporabnika, poleg tega pa je v predlogi že vključena podatkovna baza, ki se sama ustvari s potrebnimi tabelami za delovanje spletnega strežnika.

5.2.1 Avtentikacija z žetonom

Obstaja več vrst avtentikacije uporabnika, dva primera poleg avtentikacije z žetonom pa sta še:

- Avtentikacija z Windows uporabniškim računom: Če bi aplikacija delovala preko brskalnika, uporabniki pa bi uporabljali Windows operacijski sistem in bi bili prijavljeni v isti domeni (recimo uporabniki določenega podjetja), bi lahko uporabil Windows avtentikacijo. Ta je enostavna za implementacijo ter varna za uporabo, predvsem pa nemoteča za uporabnika, saj se stran avtenticira avtomatsko, brez uporabniškega posredovanja.

- Avtentikacija z piškotki: Največ se jo uporablja pri spletnih portalih, ki potrebujejo avtentikacijo uporabnika. V primeru mobilnih aplikacij, pa ta tip avtentikacije ni najboljši, saj za razliko od brskalnikov, ki poleg zahtevka na spletni strežnik sami pošljejo tudi pripadajoči piškotek, mobilne aplikacije tega ne počenjajo, če programer sam ne doda te funkcionalnosti. Druga stvar je, da mora klient, kot tudi spletni strežnik, hraniti podatke o seji. To pa pri avtentikaciji z žetonom ni potrebno.

Ker je http protokol brez stanj, pomeni, da bi mogli ob vsakem zahtevku uporabnika avtenticirati, sicer naš spletni strežnik nebi vedel komu pripada naslednji zahtevek. To nekatere vrste avtenticiranja rešijo tako, da se zapomnijo uporabnika ki se je že avtenticiral. Ker pa se ti podatki shranjuje v strežniku, in se jih lahko nabere veliko, so se pojavile težave. Največja težava je skalabilnost, saj se vse shranjuje nekje v pomnilniku.

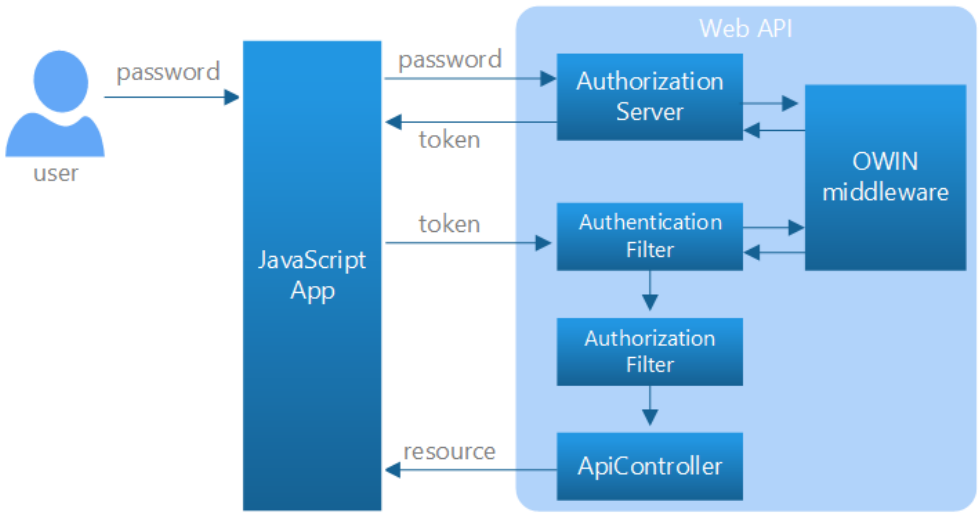
Pri avtentikaciji z žetonom pa ne shranjujemo podatkov na strežniškem delu, ampak za vsak zahtevek pošljemo zraven še žeton, ki vsebuje potrebne informacije za identifikacijo uporabnika (uporabniško ime). Ta komunikacija pa mora obvezno teči preko šifrirane HTTPS povezave. Največje prednosti takega načina avtentikacije so:

- Skalabilnost: Ker se žeton shranjuje pri uporabniku, na spletnem strežniku pa se generira, ni nobenega problema dodati še en strežnik ali pa skrbeti da strežniki niso preobremenjeni. Ni namreč pomembno na kater strežnik aplikacija pošlje zahtevek, saj žeton vsebuje dovolj podatkov za identifikacijo uporabnika

- Na več platformah: Ko načrtujemo aplikacijo, lahko nimamo pred sabo ideje kam vse se bo razširila in na katerih napravah bomo morali zagotavljati delovanje. Ampak v primeru avtentikacije z žetonom to ni problem, saj je žeton generiran s strani strežnika, API pa poskrbi za to da se uporabnik z žetonom avtenticira. Kar mora storiti aplikacija je, da v spletni strežnik pri vsakem zahtevku pošlje tudi žeton.

- Varnost: Ker za razliko od piškotkov brskalnik žetona, ki ga moramo poslati ob vsakem zahtevku, ne doda sam zraven, ampak moramo žeton dodati ročno, s tem precej omejimo napade z med spletnim ponarejanjem zahtev (CSRF). To je napad, kjer napadalec najde določen zahtevek, ki izvede neko akcijo, če je uporabnik prijavljen v tisti sistem. Recimo da klic na določen URL naslov objavi besedilo na internetni forum. Če je uporabnik prijavljen v forum, med tem pa se odpravi na napadalčevo spletno stran in klikne na zlonamerno povezavo, lahko napadalec pošlje zahtevek na točno tisti URL naslov, ki bo objavil nekaj na internetnemu forumu in ta objava se tudi zgodi, če poleg zahtevka ne pošiljamo točno določenega žetona, da lahko to preverimo na spletnem strežniku.

Prijava v sistem in s tem pridobivanje žetona poteka tako, da uporabnik v aplikacijo vnese uporabniško ime in geslo, nato pa se izvede zahtevek v avtorizacijski strežnik, ki preveri podatke v OWIN [41] vmesni programski opremi, ki ob uspešno izvedeni operaciji vrne žeton nazaj avtorizacijskemu strežniku, ta pa nazaj uporabniku. Ko ima uporabnik žeton in izvede klic, gre zahteva najprej v avtentikacijski filter, ki sprejme žeton in ga preveri v OWIN vmesni programski opremi. To se zgodi vedno, če je pogoj za izvedbo akcije to, da je uporabnik avtenticiran. Ko OWIN potrdi identiteto uporabnika, se uporabnik še avtorizira in če je tudi ta akcija uspešno izvedena, potem server vrne rezultat zahteve. Celoten postopek pa prikazuje tudi Slika 19.



Slika 19: Prikaz pridobivanja žetona in klic določene akcije [42]

5.2.2 Podatkovna baza

Podatkovno bazo sem naredil z uporabo Entity Frameworka, ki zelo poenostavi delo z podatkovno bazo in podatki. V osnovi sem naredil dva razreda, kjer vsak razred predstavlja

eno tabelo v podatkovni bazi in še en razred, kjer deklariram podatkovno bazo. V ta razred podam *connectionString*, ki ga pa preberem iz datoteke Web.config.

```
0 references | 1 request, 2 live | 0 exceptions
public async Task<TrackData> Get(int id)
{
    TrackData td = await db.TrackData
        .Where(d => d.ID == id)
        .FirstOrDefaultAsync();
    return td;
}
```

Slika 20: Primer uporabe podatkovne baze

Zgornja Slika 20 prikazuje metodo, v kateri iz tabele *TrackData* pridobimo podatke za en trening, ki ima določen ID. Kot se vidi na sliki, pri delu z podatkovno bazo uporabljam asinhrone klice. Operacije nad podatkovno bazo se sicer izvajajo enako hitro, a vendar asinhroni klic ne blokira niti in dovoli nadaljnje izvajanje, medtem ko sinhroni klic ustavi izvajanje niti.

5.3 Razvoj

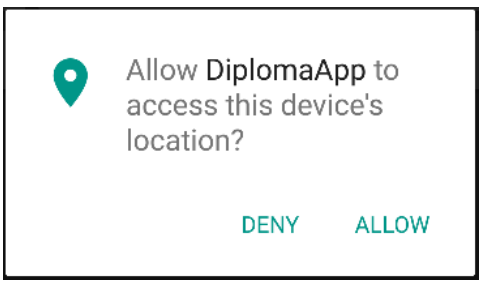
Za mobilno platformo sem izbral operacijski sistem Android, razvijal pa sem v programskem okolju Visual Studio z Xamarain vtičnikom.

Slika 21: Uradni logotip podjetja Xamarain [43]

Xamarain (Slika 21) sem izbral zato, ker mi omogoča programiranja za sistem Android v jeziku C# ter z razvojnim okoljem Visual Studio, ki je po mojem mnenju eno boljših razvojnih okolij. Velika prednost je tudi, če bi se odločil aplikacijo razviti še za iOS ter Windows Phone operacijski sistem, da lahko enako kodo uporabim tudi za ostala dva operacijska sistema, posebej naredim le grafični del. Da bom razvil aplikacijo za Android (Slika 22) operacijski sistem pa sem se odločil zato, ker je to trenutno najbolj razširjena mobilna platforma.

Slika 22: Logotip mobilnega operacijskega sistem Android [44]

Razvijal sem za verzijo API Level 23, kar je Android verzija 6.0. S to verzijo pa je Android prinesel tudi spremembe pri načinu dostopa do fizičnih senzorjev mobilnega telefona oz. pravic dostopa do njih. V moji aplikaciji sem v *Android Manifest* datoteko moral dodati dve pravici. Obe se uporablja za dostop do lokacije, ki jo mora aplikacija potrebuje. Ko uporabnik prvič zažene aplikacijo in hoče beležiti pozicijo, pa ga aplikacija vpraša kot prikazuje Slika 23, če dovoli dostop do lokacije. Če uporabnik omogoči, aplikacija deluje, sicer ne.



Slika 23: Napis, ki se prikaže uporabniku

Ko pa uporabnik enkrat že potrdi pravice za dostop, pa ga aplikacije ne sprašuje več, če le teh pravic ne prekliče iz nastavitev operacijskega sistema.

Ko sem aplikacijo dokončal, sem ugotovil, da aplikacija vsebuje varno prijavo v sistem ter prenos podatkov preko šifrirane povezave. To je dobro, saj HTTPS povezava otežuje napad vrinjenega napadalca. Ta napad deluje na princip, da se napadalec vrine med strežnikom ter odjemalcem (v mojem primeru aplikacijo), tako da gre ves promet čez njega in ga ima možnost tudi brati in ponarejati. Ker mora vsak zahtevek vsebovati tudi žeton za avtentikacijo uporabnika, je kot sem že zgoraj omenil tudi otežen napad z uporabo napada z med spletnim ponarejanjem zahtev. Pri registraciji pa preverjam tudi kvaliteto gesel, kar otežuje morebiti napad z grobo silo.

Seveda tudi moja rešitev ni popolnoma varna, a prikazuje da se lahko določeno aplikacijo z zelo malo sprememb napravi bolj varno. Je pa nekaj stvari, ki jih je tudi pri moji aplikaciji pomembno upoštevati in izboljšati. Trenutno aplikacija deluje le na mojem osebnem računalniku in je spletni strežnik podpisan z samo izdanim SSL certifikatom, aplikacija pa mu vseeno zaupa, kar ni dobro. Prva stvar je, da bi spletni strežnik rabil pravi SSL certifikat. Druga stvar, ki bi jo lahko izboljšal glede varnosti pa je, da lahko nekdo v nedogled pošilja zahtevke na spletni strežnik, kar lahko porabi vse vire in zahtevki ostalih uporabnikov nebi delovali. To bi lahko rešil z omejitvijo zahtevkov glede na interneti naslov naprave. Poleg varnosti pa bi se jo definitivno dalo izboljšati tudi v dizajnu in dodatnih funkcijah – kot je na primer prikaz zemljevida ali pa deljenja treninga na socialna omrežja, kot je po novem moderno. Vseeno menim, da bi bila varnost moje aplikacije ob zgoraj naštetih popravkih čisto primerna za njeno delovanje, saj bi večja stopnja zaščite – kot na primer prijava uporabnika z certifikatom, precej otežila postopek prijave. Zaradi tega predvidevam, da bi jo uporabljalo veliko manj uporabnikov, tisti ki bi pa jo, bi pa morali imeti tudi svoj certifikat, kar pa zelo zmanjša število potencialnih uporabnikov.

6. Sklepne ugotovitve

V diplomski nalogi sem pregledal veliko literature o internetu stvari, njegovi uporabi ter varnosti. Predvsem sem se omejil na znane vrste napadov na internet stvari ter opisal kako so jih proizvajalci opreme rešili – če so jih. Pregledal sem nekaj tehnik, kako se lahko pri načrtovanju sistema, ki bo deloval v internetu stvari čimbolj zaščitimo, na koncu pa sem z aplikacijo pokazal primer dokaj varne prijave in registracije preko mobilnega telefona na spletni strežnik.

Varnost interneta stvari je zelo pomembna, saj nas danes na vsakem koraku spremljajo naprave, ki so lahko vir veliko pomembnih in nepomembnih informacij. Videli smo nekaj primerov, kaj se lahko zgodi če dostop do take naprave dobi nepooblaščena oseba, videli pa smo tudi kako se lahko vsaj poskusimo zaščititi pred tem – sploh če smo proizvajalec opreme. To pa prav tako velja za mobilne aplikacije. Tak sistem, kot ga je uvedel Android z verzijo 6 je sicer precej dober, saj uporabniku vedno preden aplikacija prvič zahteva dostop do senzorjev sporoči in mu da možnost izbire. Medtem pa ko je v prejšnjih verzijah operacijski sistem Android o pravicah, ki jih je zahtevala aplikacija uporabnika obvestil le na začetku, ko je aplikacijo namestil. Takrat smo lahko uporabniki potrdili vse pravice naenkrat, ali pa aplikacije nismo namestili. Seveda je normalno, da aplikacija za snemanje dostopa do mikrofona, ni pa toliko samoumevno, če to želi budilka.

Po prebrani literaturi in razviti aplikaciji, ki teče na mobilnem telefonu, bi po mojem mnenju lahko internet stvari naredili bolj varen, če bi upoštevali principe, ki jih ponuja organizacija OWASP, pa tudi s tem, da bi vso povezavo šifrirali z uporabo HTTPS protokola. Na primeru moje aplikacije sem videl, da to ni zelo zahtevna naloga, vendar ogromno pripomore k varnosti celotnega sistema.

Aplikacije se razvijajo, tudi sistem kako se jih posodablja na mobilnih telefonih je zelo dobro zastavljen, saj je aplikacija na uporabnikovem telefonu skoraj vedno zadnje verzije. Tako da tudi če se odkrije kakšno varnostno pomanjkljivost, se jo lažje zakrpa kot pri fizični napravi interneta stvari. Tam pa ko je enkrat naprava narejena in če ne upošteva smernic razvoja, je ni več mogoče enostavno zakrpati. In to ni dobro, to si želimo spremeniti na bolje.

7. Literatura

[1] „ZDNet,“ 19 April 2016. [Elektronski]. Available: http://www.zdnet.com/article/hyundai-cisco-come-together-to-create-computers-on-wheels/. [Poskus dostopa 2017].

[2] „Washington Examiner,“ 15 9 2015. [Elektronski]. Available: http://www.washingtonexaminer.com/fbi-reminds-us-that-everything-can-be-hacked/article/2572021. [Poskus dostopa 2017].

[3] „Google Trends,“ [Elektronski]. Available: https://trends.google.com/trends/explore?date=all&q=internet%20of%20things,IOT. [Poskus dostopa 3 2017].

[4] „Living Internet - Internet toaster,“ 2000. [Elektronski]. Available: http://www.livinginternet.com/i/ia_myths_toast.htm.

[5] „Wikipedia - Samuel Morse,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Samuel_Morse. [Poskus dostopa 2017].

[6] „tfcbooks - Nikola Tesla interview,“ [Elektronski]. Available: http://www.tfcbooks.com/tesla/1926-01-30.htm. [Poskus dostopa 2017].

[7] „Wikipedia,“ [Elektronski]. Available: https://sl.wikipedia.org/wiki/Arpanet. [Poskus dostopa 2017].

[8] „Internet of things recruiting - The history of IoT,“ [Elektronski]. Available: http://internetofthingsrecruiting.com/the-history-of-iot/. [Poskus dostopa 2017].

[9] „The internet of things,“ [Elektronski]. Available: http://www.itu.int/osg/spu/publications/internetofthings/InternetofThings_summar

y.pdf. [Poskus dostopa 2017].

[10] „Tesla - Model S,“ [Elektronski]. Available: https://www.tesla.com/models. [Poskus dostopa 2017].

[11] „Wikipedia - Internet of things,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Internet_of_things. [Poskus dostopa 2017].

[12] „Gartner's 2016 Hype Cycle,“ [Elektronski]. Available: http://www.gartner.com/newsroom/id/3412017. [Poskus dostopa 2017].

[13] „Buisiness Insider - Here's how the Internet of Things will explode by 2020,“ [Elektronski]. Available: http://www.businessinsider.com/iot-ecosystem-internet-of-things-forecasts-and-business-opportunities-2016-2. [Poskus dostopa 2017].

[14] „Taxonomy of IoT Usecases,“ [Elektronski]. Available: https://iwringer.wordpress.com/2015/10/08/taxonomy-of-iot-usecases-seeing-iot-forest-from-the-trees/. [Poskus dostopa 2017].

[15] „Business insider - Why IoT, big data & smart farming are the future of agriculture,“ [Elektronski]. Available: http://www.businessinsider.com/internet-of-things-smart-agriculture-2016-10. [Poskus dostopa 2017].

[16] „How IoT Medical Devices Are Changing Health Care Today,“ [Elektronski]. Available: https://www.link-labs.com/blog/iot-healthcare.

[17] „Wikipedia - Smart City Amsterdam,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Smart_city#Amsterdam.

[18] „PC Mag - Here's How Hackers Steal Fingerprints From Your Phone,“ 11 8 2015. [Elektronski]. Available: http://www.pcmag.com/article2/0,2817,2489262,00.asp. [Poskus dostopa 2017].

[19] „Tech Worm - Smartwatches vulnerable to hacking,“ [Elektronski]. Available: https://www.techworm.net/2015/09/smartwatches-vulnerable-to-hacking-says-researchers.html.

[20] „PDF - Deep Spying,“ [Elektronski]. Available:

https://arxiv.org/pdf/1512.05616v1.pdf.

[21] „IT News - Hacked terminals capable of causing pacemaker deaths,“ [Elektronski]. Available: https://www.itnews.com.au/news/hacked-terminals-capable-of-causing-pacemaker-mass-murder-319508.

[22] „FDA - Managing Medical Device Cybersecurity in the Postmarket,“ [Elektronski]. Available: https://blogs.fda.gov/fdavoice/index.php/2016/12/managing-medical-device-cybersecurity-in-the-postmarket-at-the-crossroads-of-cyber-safety-and-advancing-technology/], .

[23] „ARS Technica - Car hacks could turn commutes into a scene from Speed,“ [Elektronski]. Available: https://arstechnica.com/security/2010/05/car-hacks-could-turn-commutes-into-a-scene-from-speed/.

[24] „Wikipedia - Electronic control unit,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Electronic_control_unit.

[25] „Wired,“ [Elektronski]. Available: https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/.

[26] „Ars Technica - Baby monitors,“ [Elektronski]. Available: https://arstechnica.com/security/2015/09/9-baby-monitors-wide-open-to-hacks-that-expose-users-most-private-moments.

[27] „Wikipedia - Mirai,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Mirai_(malware).

[28] „Mirai on routers,“ [Elektronski]. Available: http://www.zdnet.com/article/mirai-botnet-attack-hits-thousands-of-home-routers-throwing-users-offline/. [Poskus dostopa 2017].

[29] „Wired - The Internet of Things Is Wildly Insecure — And Often Unpatchable,“ [Elektronski]. Available: https://www.wired.com/2014/01/theres-no-good-way-to-patch-the-internet-of-things-and-thats-a-huge-problem.

[30] „Statistika - V internet povezanih naprav,“ [Elektronski]. Available:

http://www.internetlivestats.com/internet-users/.

[31] „Windows automatic updates,“ [Elektronski]. Available: https://support.microsoft.com/en-us/help/306525/how-to-configure-and-use-automatic-updates-in-windows.

[32] „Defcon - How to Hack Millions of Routers,“ [Elektronski]. Available: https://www.defcon.org/images/defcon-18/dc-18-presentations/Heffner/DEFCON-18-Heffner-Routers.pdf.

[33] „Wikipedia - DNS rebinding,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/DNS_rebinding.

[34] „Broadcomm - Products,“ [Elektronski]. Available: https://www.broadcom.com/products/.

[35] „ODM,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Original_design_manufacturer.

[36] „Wikipedia - iPhone 4,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/IPhone_4.

[37] „iPhone 4 hack,“ [Elektronski]. Available: http://blog.binamuse.com/2014/09/coregraphics-memory-corruption.html.

[38] „engineeringonline.syr.edu - Certified Security by Design: Securing Devices in the IoT Era,“ [Elektronski]. Available: https://engineeringonline.syr.edu/blog/internet-of-things-security/.

[39] „OWASP - Security by Design Principles,“ [Elektronski]. Available: https://www.owasp.org/index.php/Security_by_Design_Principles.

[40] „Wikipedia - Information security Key concepts,“ [Elektronski]. Available: https://en.wikipedia.org/wiki/Information_security#Key_concepts.

[41] „OWIN,“ [Elektronski]. Available: http://owin.org/.

[42] „Microsoft - WebApi,“ [Elektronski]. Available: https://docs.microsoft.com/en-us/aspnet/web-api/overview/security/individual-accounts-in-web-api. [Poskus

dostopa 2017].

[43] „Xamarin - logotip,“ [Elektronski]. Available: https://www.xamarin.com/branding. [Poskus dostopa 2017].

[44] „Android - Logotip,“ [Elektronski]. Available: https://developer.android.com/distribute/tools/promote/brand.html. [Poskus dostopa 2017].